

日本国特許庁

PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application:

2000年 3月31日

出願番号  
Application Number:

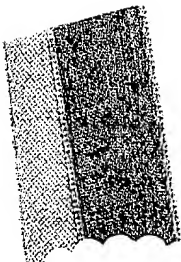
特願2000-09987

出願人  
Applicant(s):

株式会社東芝



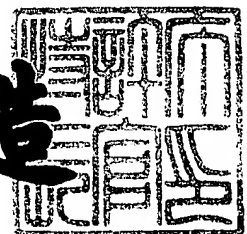
CERTIFIED COPY OF  
PRIORITY DOCUMENT



2000年 9月29日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3080530

【書類名】 特許願

【整理番号】 A000001460

【提出日】 平成12年 3月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/03

【発明の名称】 画像の輪郭抽出方法、画像からの物体抽出方法およびこの物体抽出方法を用いた画像伝送システム

【請求項の数】 10

【発明者】

    【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

    【氏名】 井田 孝

【発明者】

    【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

    【氏名】 三本杉 陽子

【発明者】

    【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

    【氏名】 堀 修

【特許出願人】

    【識別番号】 000003078

    【氏名又は名称】 株式会社 東芝

【代理人】

    【識別番号】 100058479

    【弁理士】

    【氏名又は名称】 鈴江 武彦

    【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【先の出願に基づく優先権主張】

【出願番号】 平成11年特許願第301415号

【出願日】 平成11年10月22日

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705037

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像の輪郭抽出方法、画像からの物体抽出方法およびこの物体抽出方法を用いた画像伝送システム

【特許請求の範囲】

【請求項 1】

輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第 2 のステップと、シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、各々の前記相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第 3 のステップとから成り、前記第 3 のステップを所定の回数繰り返すことにより、シェイプデータの輪郭線を前記物体の輪郭線に一致させる輪郭抽出処理方法において、

前記輪郭抽出処理を所定の回数反復して行い、

その反復の初めには前記画像データとシェイプデータを縮小して輪郭抽出処理を行うことを特徴とする画像の輪郭抽出方法。

【請求項 2】

前記輪郭抽出処理は、輪郭抽出処理の反復回数が進むにつれて前記探索基準ブロックのサイズは小さくすることを特徴とする請求項 1 記載の画像の輪郭抽出方法。

【請求項 3】

輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと

、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第2のステップと、シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、各々の前記相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第3のステップとから成り、前記第3のステップを所定の回数繰り返すことにより、シェイプデータの輪郭線を前記物体の輪郭線に一致させる輪郭抽出処理方法において、

前記第2のステップでの処理は、相似ブロックの探索範囲を探索基準ブロック内のシェイプデータの輪郭の方向と垂直な方向に制限することを特徴とする画像の輪郭抽出方法。

#### 【請求項4】

抽出対象物体が撮像された画像データの各部分を同じ画像データの別の部分または別の画像データと比較することにより、この画像データ内の抽出対象物体の暫定的な領域を表わす画像であるシェイプデータを生成する第1のステップと、

前記画像データと暫定的に生成した前記シェイプデータとを用いて、前記シェイプデータの輪郭を前記抽出対象物体の輪郭に一致させる第2のステップとから成ることを特徴とする画像からの物体抽出方法。

#### 【請求項5】

前記第1のステップは、予め複数の候補領域を決めておき、これら候補領域を選択的に画像データ内に設定した場合に、前記候補領域の内側と外側の画素値の統計的な性質が最も隔たるような候補領域を前記抽出対象物体の暫定的な領域として設定することを含むことを特徴とする請求項4記載の画像からの物体抽出方法。

#### 【請求項6】

前記第1のステップは、現入力画像データと、予め取得しておく前記抽出対象物体が写っていない背景画像データとを比較し、現入力画像データと背景画像データの値が異なる領域を前記抽出対象物体の暫定的な領域として設定することを含むことを特徴とする請求項4記載の画像からの物体抽出方法。

【請求項 7】

前記第 1 のステップは、既に抽出対象物体の領域が求められたフレームの画像データとシェイプデータをそれぞれ基準画像データ、基準シェイプデータとして保持し、現入力画像データに対して、前記基準画像データを参照して動き検出を行い、その動き検出結果に基づいて前記基準シェイプデータに動き補償を行うことで前記暫定的なシェイプデータを生成することを含むことを特徴とする請求項 1 記載の画像からの物体抽出方法。

【請求項 8】

抽出対象物体が撮像された画像データと、

前記画像データ内の抽出対象物体の暫定的な領域を表す画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心部分において所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと、

各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第 2 のステップと、

前記シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、前記相似ブロックより得た縮小処理によるサイズ補正済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第 3 のステップとから成る物体抽出方法において、

前記シェイプデータは、互いに異なる物体の領域と背景領域ではそれぞれ互いに異なる画素値を有し、

前記第 3 のステップにおける縮小処理では、前記シェイプデータのサンプリング点の周囲のいずれかの画素の画素値をサンプリング値とすることを特徴とする画像からの物体抽出方法。

【請求項 9】

送受信データを相互に送受するサーバーとクライアントからなる伝送システムにおいて、

前記クライアントは、

抽出対象物体が撮像された画像データから物体を抽出して抽出画像データを得る物体抽出手段と、

前記抽出画像データをそのままか、あるいは圧縮し、上り送受信データとして前記サーバーに送るクライアント送信手段と、

前記サーバーから送られてくる下り送受信データを受信して、その送受信データが圧縮されていない場合にはそのまま画像データとし、圧縮されている場合には画像データを再生するクライアント受信手段と、

前記画像データを表示する手段とで構成され、

前記サーバーは、

前記上り送受信データを受信して、その送受信データが圧縮されていない場合にはそのまま抽出画像データとし、圧縮されている場合には抽出画像データを再生するサーバー受信手段と、

前記抽出画像データを一つの合成画像データに合成する合成手段と、

前記合成画像データをそのままか、あるいは圧縮し、前記下り送受信データとして送信するサーバー送信手段とで構成される、ことを特徴とする伝送システム

【請求項 1 0】

送受信データを相互に送受する複数の通信端末からなる伝送システムにおいて

前記通信端末は、

抽出対象物体が撮像された画像データから物体を抽出して抽出画像データを得る物体抽出手段と、

前記抽出画像データをそのままか、あるいは圧縮し、通信端末間送受信データとして別の通信端末に送る送信手段と、

別の通信端末から送られてくる前記通信端末間送受信データを受信して、その送受信データが圧縮されていない場合にはそのまま抽出画像データとし、圧縮されている場合には抽出画像データを再生する受信手段と、

前記抽出画像データを一つの合成画像データに合成する合成手段と、

前記合成画像データを表示する手段とにより構成される、ことを特徴とする伝



送システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像から輪郭抽出対象物体の領域を抽出するための画像の輪郭抽出方法、画像からの物体抽出方法およびこの物体抽出方法を用いた画像伝送システムに関する。

【0002】

【従来の技術】

画像から目的の被写体の領域を抽出する技術は、例えば、背景を別の画像に置き換えるといった加工などに有用である。その際、目的の被写体である抽出対象物体の領域が正しく求まっていないと、高い品質の加工画像は得られない。

【0003】

そのため、何らかの方法で物体の概略形状が求まっているときに、その概略形状を元にして、より高い精度で物体領域を求める技術が求められており、その手法の一つとして、本件発明者らは文献：「自己相似写像による輪郭線のフィッティング」（井田、三本杉、第5回画像センシングシンポジウム講演論文集、C-15、pp.115-120、June 1999.）に開示した如きものを提案した。この技術は、自己相似写像を用いる手法であって、その概要を図5～図8を用いて説明しておく。

【0004】

この技術は、図5に示すように、処理対象のフレーム画像中における抽出対象物体（正しい物体領域）1の画像部分を斜線で、また、抽出対象物体の概略形状2を太線で表わしている。抽出対象物体の概略形状2は例えば、オペレータが画面に表示されたフレーム画像上において抽出対象物体の外形に沿うようにマニュアル操作で大雑把に描いた外形線であり、抽出対象物体1の輪郭1aとはズレがある。

【0005】

そこで、この抽出対象物体の概略形状2の内側の画素に画素値“255”を代

入し、そして、外側の画素に画素値“0”を代入する。すなわち、抽出対象物体の概略形状2の内側の画素全てが画素値“255”に塗りつぶされた、その背景は全て画素値“0”に塗りつぶされた画像（すなわち、二値画像）が得られる。このような処理を施して得た画像（二値画像）をシェイプデータあるいはアルファマップなどと呼ぶ。

## 【0006】

自己相似写像を用いる手法によれば、暫定的に与えた抽出対象物体の概略形状2、すなわち、言い換えれば暫定的に設定したシェイプデータの輪郭を、求めたい物体の正しい輪郭である抽出対象物体1の輪郭1aに一致させることができる。

## 【0007】

そのためには、まず、暫定的なシェイプデータ（暫定的なアルファマップ）の輪郭（抽出対象物体の概略形状2）に沿って探索基準ブロックB1, B2, ..., B<sub>n-1</sub>, B<sub>n</sub>を配置する。

## 【0008】

これは、暫定的なシェイプデータ（暫定的なアルファマップ）の画像について左上画素（座標位置0, 0）から右方向へ、上のラインから下のラインへ順にスキャンし、すなわち、基準座標位置からXYスキャンさせて画素を調べていき、左隣りあるいは上隣りの画素と画素値が異なり、それまでに設定したブロックに含まれない画素である場合に、その画素を中心にして所定の大きさ（ブロックサイズb）のブロックを配置することで行う。これにより、図5にB1, B2, ..., B<sub>n</sub>で示すようにブロックが数珠つなぎに重なり合いながら設定された探索基準ブロックが得られる。

## 【0009】

次に、処理対象のフレーム画像中における該当座標位置上にそれぞれの探索基準ブロックB1, ..., B<sub>n</sub>を位置させる。これにより、抽出対象物体（正しい物体領域）1の輪郭位置を一部に含む抽出対象物体1の部分領域をそれぞれの探索基準ブロックB1, ..., B<sub>n</sub>が含むことになるので、それぞれそのブロック内の画素状況が似ている相似ブロックを求める。

## 【 0 0 1 0 】

相似ブロックは対応する探索基準ブロックを領域拡大した範囲で試行錯誤的に各相似候補ブロックを設定し、その相似候補ブロック内の画像を縮小したものと探索基準ブロック内の画像との誤差が最小となるものを見つけてそれを相似ブロックとして求める。例えば、図 6 に、探索基準ブロック B 1 の相似ブロック B s 1 を示すが、この図の如き対応となる。

## 【 0 0 1 1 】

このようにして、それぞれの探索基準ブロック B 1, ..., B n に対しての抽出対象物体（正しい物体領域） 1 の相似ブロックを求める。

このように、相似ブロックとは、対応する探索基準ブロックよりも大きく、その画像データを縮小したときの画素値が、探索基準ブロック内の画像データとほぼ等しくなるものをいう。

## 【 0 0 1 2 】

相似ブロックを見つけるための具体的手法は、次の如きである。例えば、探索基準ブロック B 1 の相似ブロックを見つけるためには、図 7 に示したように、探索基準ブロック B 1 の周囲に適宜な大きさの探索領域 F s 1 を設定する。そして、この探索領域 F s 1 の内部において相似候補ブロック B c を様々に設定しながら、その度にブロックサイズへの縮小処理と探索基準ブロック B 1 との誤差評価を行い、誤差が最小となった相似候補ブロックを相似ブロック B c d として決定する。

## 【 0 0 1 3 】

図 7 の例では、探索基準ブロック B 1 の縦横 2 倍で中心が同じ位置である相似候補ブロック B c を基準にしてこれを上下左右に各々 w 画素の範囲で 1 画素ずつずらしながら誤差評価を行う。

## 【 0 0 1 4 】

なお、図 6 では探索基準ブロック B 1 の相似ブロック B s 1 しか示していないが、もちろん図 5 に示した全ての探索基準ブロック B 1, ..., B n に対してそれぞれ相似ブロックを求める。

## 【 0 0 1 5 】

次に、シェイプデータ（アルファマップ）に対して、各探索基準ブロック  $B_1$  ,  $\dots$   $B_n$  の位置に該当するデータについて、その探索基準ブロック対応の相似ブロック内の画像から得たシェイプデータ（アルファマップ）で置き換えると云う手法で補正する。相似ブロックは探索基準ブロックよりサイズが大きいので、当然のことながらサイズは縮小処理して合わせる（サイズの正規化）。

## 【0016】

このようにして探索基準ブロック対応の相似ブロック内の画像から得た正規化済みのシェイプデータ（アルファマップ）で置き換えると云う手法で補正する。これを全ての探索基準ブロック  $B_1$  ,  $\dots$   $B_n$  において1回ずつ行くと、シェイプデータの輪郭（抽出対象物体の概略形状2）は正しい物体領域1の輪郭1aに近づく。そして、置き換えを再帰的に反復することにより、図8に示したように、シェイプデータの輪郭（抽出対象物体の概略形状2）は正しい物体領域1の輪郭1aの近くに収束する。

## 【0017】

この手法では探索基準ブロック  $B_1$  ,  $\dots$   $B_n$  に正しい物体領域1の輪郭線が含まれている必要がある。従って、正しい物体領域1の輪郭線に対する探索基準ブロック  $B_1$  ,  $\dots$   $B_n$  のずれが大きい場合には、まず、大きなブロックを用いて上述した処理を行うことで、シェイプデータの輪郭（抽出対象物体の概略形状2）を正しい物体領域1の輪郭1aの近くに寄せ、次に、小さなブロックで上述の処理を行うことで、細かな凸凹を一致させる。

これにより、輪郭のずれが大きい場合でも高い精度で輪郭の抽出が可能となる。そのフローチャートを図9に示す。図9のフローチャートに従って処理を説明すると、まず、

【ステップS1の処理】：探索基準ブロックのブロックサイズを  $b$  とすると、このブロックサイズ  $b$  を  $A$  と置く。

## 【0018】

【ステップS2の処理】：このブロックサイズ  $b$  を用いて、先に図5～図8を用いて説明した輪郭位置補正処理を行う。

## 【0019】

【ステップ S 3 の処理】：次に、探索基準ブロックのブロックサイズ  $b$  を前回の半分の値にする。

【0 0 2 0】

【ステップ S 4 の処理】：その結果、探索基準ブロックのブロックサイズ  $b$  が  $Z (< A)$  よりも小さい場合には処理を終了し、そうでない場合にはステップ S 2 の処理に戻る。

【0 0 2 1】

このような処理により、暫定的に与えたシェイプデータ（暫定的に与えたアルファマップ）の輪郭が、目的の抽出対象の輪郭とずれが大きいような場合でも、目的の抽出対象の輪郭に一致させるように輪郭位置補正処理することができ、従って、この輪郭位置補正処理による補正済みのシェイプデータ（アルファマップ）を用いることで画像中から高い精度で当該目的の抽出対象画像の輪郭を抽出できるようになる。

【0 0 2 2】

【発明が解決しようとする課題】

画像データから目的の画像部分を抽出する方法として、何らかの方法で物体の概略形状が求まっているときに、その概略形状をもとにして、抽出対象物体の領域を求める技術が求められており、その手法の一つとして、自己相似写像を用いる手法がある。

【0 0 2 3】

この技術は、画像中の目的の画像部分の概略形状をオペレータがマニュアル操作で大雑把に描くことにより、抽出対象物体の概略形状 2 を得、この抽出対象物体の概略形状 2 の内側の画素に画素値 “2 5 5” を代入し、そして、外側の画素に画素値 “0” を代入する処理（つまり、二値化）を施すことで、シェイプデータを得る。

【0 0 2 4】

そして、シェイプデータの輪郭（抽出対象物体の概略形状 2）に沿って探索基準ブロック  $B_1, B_2, \dots, B_{n-1}, B_n$  を配置する。

【0 0 2 5】

次に、それぞれの探索基準ブロック  $B_1, \dots, B_n$  についてそれぞれ相似ブロックを求める。

【0026】

次に、シェイプデータについて、各探索基準ブロック  $B_1, \dots, B_n$  の位置のデータを、その探索基準ブロック対応の相似ブロックにて切り出して縮小したもので置き換える。これを全ての探索基準ブロック  $B_1, \dots, B_n$  において1回ずつ行くと、シェイプデータの輪郭（抽出対象物体の概略形状2）は正しい物体領域1の輪郭1aに近づく。そして、置き換えを再帰的に反復することにより、図8に示したように、シェイプデータの輪郭（抽出対象物体の概略形状2）は正しい物体領域1の輪郭1aの近くに収束することになる。それは、各相似ブロックが、それぞれが対応している探索基準ブロックよりも大きく、その画像データを縮小したときの画素値が、探索基準ブロック内の画像データとほぼ等しくなるものだからである。

【0027】

この手法では探索基準ブロック  $B_1, \dots, B_n$  に正しい物体領域1の輪郭線が含まれている必要がある。従って、正しい物体領域1の輪郭線に対する探索基準ブロック  $B_1, \dots, B_n$  のずれが大きい場合には、まず、大きなブロックを用いて上述した処理を行うことで、暫定的に与えたシェイプデータの輪郭（抽出対象物体の概略形状2）を正しい物体領域1の輪郭1aの近くに寄せ、次に、小さなブロックで上述の処理を行うことで、細かな凸凹までもを一致させる補正ができる。そして、これにより、輪郭のずれが大きい場合でも高い精度で輪郭の抽出が可能となる。

【0028】

自己相似写像を用いるこのような手法によれば、暫定的に与えた抽出対象物体の概略形状2、すなわち、言い換えれば暫定的に与えたシェイプデータの輪郭を、正しい物体領域1の輪郭1aに一致させることができる。

【0029】

しかしながら、この手法の場合、相似ブロック探索の演算量が多く、画像から目的画像部分を高速で抽出することが困難であるという問題があった。そのため

、動画像などに適用するにはさらなる改善が必要である。

【 0 0 3 0 】

そこで、この発明の目的とするところは、処理量を削減でき、画像から目的画像部分を高速で抽出することが可能となる画像の輪郭抽出方法を提供することにある。

【 0 0 3 1 】

【課題を解決するための手段】

上記目的を達成するため、本発明は次のように構成する。

【 0 0 3 2 】

〔 1 〕 第 1 には、輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第 2 のステップと、シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、各々の前記相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第 3 のステップとから成り、前記第 3 のステップを所定の回数繰り返すことにより、シェイプデータの輪郭線を前記物体の輪郭線に一致させる輪郭抽出処理方法において、

前記輪郭抽出処理を所定の回数反復して行い、

その反復の初めには前記画像データとシェイプデータを縮小して輪郭抽出処理を行うことを特徴とする。

【 0 0 3 3 】

本発明においては、自己写像を用いる方法を採用して暫定のシェイプデータ（アルファマップ）を輪郭抽出対象物体の輪郭に一致するように修正するが、そのために輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデー

タを用い、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する。そして、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する。そして、シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、各々の前記相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることにより、シェイプデータの補正処理をする。

## 【 0 0 3 4 】

本発明では、前記輪郭抽出処理を所定の回数反復して行い、その反復の初めには前記画像データとシェイプデータを縮小して輪郭抽出処理を行う。

## 【 0 0 3 5 】

すなわち、複数回繰り返される輪郭抽出処理においては、そのはじめの方では前記画像データとシェイプデータを縮小して輪郭抽出処理を行うが、処理対象の画像サイズを小さくすれば、それに合わせて輪郭のずれの画素数も小さくなることから、探索基準ブロックのブロックサイズをそれほど大きくしなくてもシェイプデータの輪郭を正しい位置に近づけることができることを利用している。故に、本発明では、はじめの方の回の輪郭抽出処理においては、画像データは縮小したものをを用いると共に、あわせてシェイプデータおよび探索基準ブロックも縮小したものをを用いて輪郭抽出処理を行う。

## 【 0 0 3 6 】

ここでの処理に従来よりも小さなサイズのブロックで処理が行えるのでその分、演算量も少なく済む。輪郭抽出処理の回数が進むと、元の画像サイズでの輪郭抽出処理が行われることで、最終的には輪郭抽出対象物体の持つ輪郭の細かい凸凹も反映されるようにシェイプデータの補正がなされ、この補正されたシェイプデータを用いることで、輪郭抽出対象物体の輪郭を精度良く抽出できるようになる。

## 【 0 0 3 7 】

従って、この発明によれば、シェイプデータ（アルファマップ画像）の輪郭位



置補正の処理量を軽減でき、画像から目的画像部分を高速で抽出することが可能となる画像の輪郭抽出方法を提供できる。

## 【 0 0 3 8 】

〔 2 〕 また、本発明は、輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第 2 のステップと、シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、各々の前記相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第 3 のステップとから成り、前記第 3 のステップを所定の回数繰り返すことにより、シェイプデータの輪郭線を前記物体の輪郭線に一致させる輪郭抽出処理方法において、

前記第 2 のステップでの処理は、相似ブロックの探索範囲を探索基準ブロック内のシェイプデータの輪郭の方向と垂直な方向に制限することを特徴とする。

## 【 0 0 3 9 】

本発明においては、自己写像を用いる方法を採用してシェイプデータ（アルファマップ）を輪郭抽出対象物体の輪郭に一致するように修正するが、そのために輪郭抽出対象物体が撮像された画像データと、この画像データ内の輪郭抽出対象物体の領域を表わす暫定的な輪郭形状の二値化画像であるシェイプデータを用い、前記シェイプデータの輪郭部分に、それぞれ中心位置を置いて所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する。各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する。シェイプデータのうち、各探索基準ブロック内におけるシェイプデータを、各々の相似ブロックより得た縮小処理によるサイズ調整済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする。

## 【 0 0 4 0 】

本発明では、前記第 2 のステップでの処理、すなわち、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する相似ブロック探索処理は、相似ブロックの探索範囲を探索基準ブロック内のシェイプデータの輪郭の方向と垂直な方向に制限するようにする。

## 【 0 0 4 1 】

すなわち、ある探索基準ブロックに対する相似ブロック探索処理は、従来手法によれば、その探索基準ブロックを中心に、上下左右所定の範囲内でブロックサイズを拡張して類似の画素パターンを示すものを探すことで実施するが、本発明では相似ブロックの探索範囲を探索基準ブロック内のシェイプデータの輪郭の方向と垂直な方向に制限する。これにより、演算量を軽減する。

## 【 0 0 4 2 】

輪郭抽出対象物体の輪郭は、もちろん未知であり、暫定的に与えたシェイプデータの輪郭をどちらに動かせば輪郭抽出対象物体の輪郭に近づくのかも、相似ブロック探索時には当然分からないわけであるが、シェイプデータの輪郭の方向と輪郭抽出対象物体の輪郭の方向は経験的にみて、ほぼ合っていることがほとんどなので、シェイプデータの輪郭の方向と垂直な方向に対して探索するのが最も合理的である。

## 【 0 0 4 3 】

従って、この発明によれば、シェイプデータ（アルファマップ）の輪郭位置補正の処理量を軽減でき、画像から目的画像部分を高速で抽出することが可能となる画像の輪郭抽出方法を提供できる。

## 【 0 0 4 4 】

更に、本発明は、抽出対象物体が撮像された画像データの各部分を同じ画像データの別の部分または別の画像データと比較することにより、この画像データ内の抽出対象物体の暫定的な領域を表わす画像であるシェイプデータを生成する第 1 のステップと、前記画像データと暫定的に生成した前記シェイプデータとを用いて、前記シェイプデータの輪郭を前記抽出対象物体の輪郭に一致させる第 2 の

ステップとから成ることを特徴とする画像からの物体抽出方法を提供する。

【 0 0 4 5 】

本発明によると、物体を所定位置に固定しなくとも適正に物体を抽出できる。

【 0 0 4 6 】

本発明は、抽出対象物体が撮像された画像データと、前記画像データ内の抽出対象物体の暫定的な領域を表す画像であるシェイプデータを入力とし、前記シェイプデータの輪郭部分に、それぞれ中心部分において所定サイズの探索基準ブロックを、互いの位置をずらして複数設定する第 1 のステップと、各探索基準ブロック毎に、そのブロック内での前記画像データの図柄が相似で、かつ、前記探索基準ブロックよりも大きいブロックサイズの相似ブロックを同じ画像の中から探索する第 2 のステップと、前記シェイプデータのうち、前記各探索基準ブロック内におけるシェイプデータを、前記相似ブロックより得た縮小処理によるサイズ補正済みの補正用シェイプデータと置き換えることによりシェイプデータの補正処理をする第 3 のステップとから成る物体抽出方法において、

前記シェイプデータは、互いに異なる物体の領域と背景領域ではそれぞれ互いに異なる画素値を有し、前記第 3 のステップにおける縮小処理では、前記シェイプデータのサンプリング点の周囲のいずれかの画素の画素値をサンプリング値とすることを特徴とする画像からの物体抽出方法を提供する。

【 0 0 4 7 】

本発明は、送受信データを相互に送受するサーバーとクライアントからなる伝送システムにおいて、前記クライアントは、抽出対象物体が撮像された画像データから物体を抽出して抽出画像データを得る物体抽出手段と、前記抽出画像データをそのままか、あるいは圧縮し、上り送受信データとして前記サーバーに送るクライアント送信手段と、前記サーバーから送られてくる下り送受信データを受信して、その送受信データが圧縮されていない場合にはそのまま画像データとし、圧縮されている場合には画像データを再生するクライアント受信手段と、前記画像データを表示する手段とで構成され、

前記サーバーは、前記上り送受信データを受信して、その送受信データが圧縮されていない場合にはそのまま抽出画像データとし、圧縮されている場合には抽

出面像データを再生するサーバー受信手段と、前記抽出画像データを一つの合成画像データに合成する合成手段と、前記合成画像データをそのままか、あるいは圧縮し、前記下り送受信データとして送信するサーバー送信手段とで構成される、ことを特徴とする伝送システムを提供する。

## 【 0 0 4 8 】

## 【発明の実施の形態】

以下、本発明の実施形態について図面を参照して説明する。本発明においては、第 1 には、画像データと暫定的なシェイプデータと探索基準ブロックを用いて画像中の輪郭抽出対象物体の輪郭抽出に用いるシェイプデータ（アルファマップ）を輪郭抽出対象物体の輪郭に一致させるべく修正する輪郭補正処理を反復して行い、その反復の初めには画像データとシェイプデータと探索基準ブロックを縮小して輪郭補正処理を行う。

## 【 0 0 4 9 】

すなわち、この第 1 の発明では、はじめの方の回では輪郭補正処理は、画像サイズを小さくして処理するようにするので、それに合わせて輪郭のずれの画素数も小さくなることから、探索基準ブロックのブロックサイズをそれほど大きくしなくてもシェイプデータ（アルファマップ）の輪郭を正しい位置に近づけることができることを利用する。このように、従来よりも小さなブロックサイズの探索基準ブロックで処理が行えるので演算量も少なくできる。そして、最後には元の画像サイズで輪郭補正処理をすることで輪郭の細かい凸凹にもフィットさせたシェイプデータ（アルファマップ）を得ることができるようになる。

## 【 0 0 5 0 】

以下、本発明の実施形態について、図面を用いて説明する。

## 【 0 0 5 1 】

## （本発明で用いる基本処理内容）

本発明においても輪郭補正処理の基本的な処理内容は従来例で説明した手法を踏襲する。

## 【 0 0 5 2 】

## 〔基本部分の手法〕

すなわち、基本部分の手法は、次のようなものである。いま、図 5 に示すように、抽出処理対象の画像が撮像されたフレーム画像（原画像）中における当該抽出対象物体（正しい物体領域）1 の画像があり、これを輪郭抽出処理することを考える。

## 【0 0 5 3】

## 〔処理 1〕

## ＜抽出対象物体の概略形状設定＞

オペレータは、原画像をディスプレイ画面に表示させ、画面上において当該抽出対象物体（正しい物体領域）1 の画像に抽出対象物体の概略形状 2 を設定する。フレーム画像（原画像）と、抽出対象物体の概略形状 2 のデータは座標位置は両方で共通化されているが、レイヤを別にするなど、重ね表示はできても、両者のデータが混在しないように管理する。

## 【0 0 5 4】

抽出対象物体の概略形状 2 は例えば、オペレータが画面に表示されたフレーム画像（原画像）上において抽出対象物体の外形に沿うようにマニュアル操作で大雑把に描いた外形線であり、抽出対象物体 1 の輪郭 1 a とはズレがある。

## 【0 0 5 5】

そこで、抽出対象物体の概略形状 2 の輪郭を、求めたい物体の正しい輪郭である抽出対象物体 1 の輪郭 1 a に一致させるための処理を、自己相似写像の手法により行う。

## 〔処理 2〕

自己相似写像法は次の通りである。

## 【0 0 5 6】

## 〔処理 2 - 1〕

## ＜シェイプデータ（アルファマップ）生成＞

まず、前記抽出対象物体の概略形状 2 のデータを対象に、当該抽出対象物体の概略形状 2 の内側の画素に画素値“2 5 5”を代入し、そして、外側の画素に画素値“0”を代入する処理を施す。これにより、抽出対象物体の概略形状 2 の内側の画素全てが画素値“2 5 5”に塗りつぶされ、また、その背景は全て画素

値“0”に塗りつぶされた画像（つまり、二値画像）であるシェイプデータ（アルファマップ）が得られる。

【0057】

[処理2-2]

＜探索基準ブロックの配置＞

次に、シェイプデータ（アルファマップ）の輪郭（抽出対象物体の概略形状2）に沿って探索基準ブロック $B_1$ ,  $B_2$ , ...,  $B_{n-1}$ ,  $B_n$ を配置する。

【0058】

これは、アルファマップの左上画素（座標位置0, 0）から右方向へ、上のラインから下のラインへ順にスキャンし、すなわち、基準座標位置からXYスキャンさせて画素を調べていき、左隣りあるいは上隣りの画素と画素値が異なり、それまでに設定したブロックに含まれない画素である場合に、その画素を中心にして所定の大きさ（ブロックサイズ $b$ ）のブロックを配置することで行う。これにより、図5に $B_1$ ,  $B_2$ , ...,  $B_n$ で示すようにそれぞれ固有の位置座標を以て配置される探索基準ブロックが得られる。この例では、探索基準ブロック $B_1$ ,  $B_2$ , ...,  $B_n$ が数珠つなぎに重なり合いながら設定されている。

【0059】

[処理2-3]

＜相似ブロックの取得＞

それぞれ固有の位置座標を持つ探索基準ブロック $B_1$ ,  $B_2$ , ...,  $B_n$ が得られたならば、次に、処理対象のフレーム画像（原画像）上にそれぞれの探索基準ブロック $B_1$ , ...,  $B_n$ を座標位置を合わせて配置させる。

これにより、それぞれの探索基準ブロック $B_1$ , ...,  $B_n$ はそれぞれの座標位置において、自己の持つブロックサイズの範囲で抽出対象物体（正しい物体領域）1の輪郭位置を一部に含む抽出対象物体1の部分領域を特定することになるので、これを用いてそれぞれそのブロック内の画素状況が似ている相似ブロックを求める。

【0060】

ここで、相似ブロックとは、対応する探索基準ブロックよりも大きく、その画

像データを縮小したときの画素値が、探索基準ブロック内の画像データとほぼ等しくなるものをいう。

#### 【0061】

相似ブロックは、対応する探索基準ブロックを領域拡大した範囲で試行錯誤的に各相似候補ブロックを設定し、そのブロックの範囲の画像について探索基準ブロックのブロックサイズに縮小したものと、対応する探索基準ブロックのブロック内画像の画素状況（簡単に言えばブロック内の図柄）が似ているか否かの度合いを誤差評価により調べ、誤差が最小（評価結果が最良）となるものを見つけてそれを相似ブロックとして求める。

#### 【0062】

このようにして、それぞれの探索基準ブロック  $B_1, \dots, B_n$  に対しての抽出対象物体（正しい物体領域）1の相似ブロックを求める。

#### 【0063】

相似ブロックを見つけるための具体的手法は、次の如きである。

例えば、探索基準ブロック  $B_1$  の相似ブロックを見つけるためには、図7に示したように、探索基準ブロック  $B_1$  の周囲に適宜な大きさの探索領域  $F_{s1}$  を設定する。そして、この探索領域  $F_{s1}$  の内部において相似候補ブロック  $B_c$  を様々に設定しながら、その度に探索基準ブロック  $B_1$  のブロックサイズへの縮小処理と、縮小処理後の相似候補ブロックと探索基準ブロック  $B_1$  との画素分布の類似度を評価する誤差評価を行い、誤差が最小（評価結果が最良）となった相似候補ブロックを相似ブロック  $B_{cd}$  として決定する。

#### 【0064】

図7の例では、探索基準ブロック  $B_1$  の縦横2倍で中心の座標位置が同位置である相似候補ブロック  $B_c$  を基準にして、これを上下左右に各々  $w$  画素の範囲で1画素ずつずらすことで得られる新たな相似候補ブロックを用い、探索基準ブロック  $B_1$  の画素分布に対する誤差評価を行う。

#### 【0065】

なお、図6では探索基準ブロック  $B_1$  の相似ブロック  $B_{s1}$  しか示していないが、もちろん図5に示した全ての探索基準ブロック  $B_1, \dots, B_n$  に対してそれぞ

れ最終的な相似ブロック  $B_{cd1}, \dots, B_{cdn}$  を求める。

【 0 0 6 6 】

[処理 2 - 4]

＜輪郭位置補正処理＞

全ての探索基準ブロック  $B_1, \dots, B_n$  に対してそれぞれ相似ブロック  $B_{cd1}, \dots, B_{cdn}$  が求められたならば、次に、シェイプデータ（アルファマップ）に対し、各探索基準ブロック  $B_1, \dots, B_n$  内の部分を、補正用のシェイプデータ（アルファマップ）で置き換えると云う処理をする。この補正用のシェイプデータ（アルファマップ）は、それぞれその探索基準ブロックに対応する相似ブロック  $B_{cd1}, \dots, B_{cdn}$  でシェイプデータ（アルファマップ）から切り出し、探索基準ブロックのブロックサイズに縮小処理したものである。

【 0 0 6 7 】

これを全ての探索基準ブロック  $B_1, \dots, B_n$  において 1 回ずつ行くと、シェイプデータの輪郭（抽出対象物体の概略形状 2）は正しい物体領域 1 の輪郭 1 a に近づく。そして、このような置き換え処理を再帰的に反復することにより、図 8 に示したように、シェイプデータの輪郭（抽出対象物体の概略形状 2）は正しい物体領域 1 の輪郭 1 a の近くに収束する。

【 0 0 6 8 】

このような処理により、シェイプデータ（アルファマップ）の輪郭は正しい物体領域 1 の持つ輪郭と細かな凸凹まで一致させることができる。

【 0 0 6 9 】

但し、この手法では「フレーム画像（原画像）上に探索基準ブロック  $B_1, \dots, B_n$  を配置した場合に、各探索基準ブロック  $B_1, \dots, B_n$  内には正しい物体領域 1 の輪郭線が含まれている」という条件が成立している必要がある。従って、正しい物体領域 1 の輪郭線に対する探索基準ブロック  $B_1, \dots, B_n$  のずれが大きい場合には、まず、大きなブロックを用いて上述した処理を行うことで、シェイプデータの輪郭（抽出対象物体の概略形状 2）を正しい物体領域 1 の輪郭 1 a の近くに寄せる処理をして必要条件を整え、次に、小さなブロックで上述の処理を再帰的に行うことで、最終的に正しい物体領域 1 の輪郭に一致させる。



## 【0070】

このような処理により得られた補正済みのシェイプデータ（アルファマップ）の輪郭は、輪郭抽出対象物体（正しい物体領域1）の持つ輪郭1aと細かな凸凹まで一致させることができる。

そして、これにより、輪郭のずれが大きい場合でも高い精度で輪郭の抽出が可能となる。

## 【0071】

以上が、シェイプデータ（アルファマップ）の輪郭位置補正処理であり、本発明においても、基本的な処理の内容は、従来からあるこのような自己相似写像法を踏襲している。但し、本発明では、演算処理の負担を軽減させるために、輪郭位置補正処理で用いる探索基準ブロックのブロックサイズを従来手法より小さくするようにしている。手法の詳細を説明する。

## 【0072】

## （第1の実施形態）

すなわち、従来手法においては、図9に示したように、シェイプデータ（アルファマップ）も、また、原画像（抽出対象の撮像されているフレーム画像）も、また、探索基準ブロックも元々のサイズのままのもの（縮小していないもの）を用いた。しかし、それを本発明では演算処理低減のために、サイズを縦横それぞれ今までの半分のサイズにして処理を進めるようにする。

## 【0073】

すなわち、従来は、探索基準ブロックの一辺あたりのブロックサイズを $b$ とすると、この $b$ は $A$ と置いた。そして、この $A$ なるブロックサイズ $b$ の探索基準ブロックを用い、かつ、原画像（抽出対象の撮像されているフレーム画像）およびシェイプデータ（アルファマップ）も元々のサイズの画像を用い、先に図5～図8を用いて説明した輪郭位置補正処理を行った。そして、次に、 $A$ なるブロックサイズ $b$ を、その半分の値にし、ブロックサイズ $b$ が $Z$ （ $<A$ ）よりも小さい場合には終了し、そうでない場合にはブロックサイズをさらに半分にしてこれを用いて輪郭位置補正処理を行うと云う処理を繰り返した。

## 【0074】

そして、このような処理により、輪郭のずれが大きい場合でも高い精度で輪郭抽出できるようにした。

#### 【 0 0 7 5 】

これに対して、本発明では、輪郭位置補正処理は粗調整段階と、本調整段階との2段階処理とし、粗調整段階での輪郭位置補正処理では、はじめから暫定的なシェイプデータ（アルファマップ）を縦横1/2に縮小したものを、また、原画像も縦横1/2に縮小した画像を用いるようにし、それに合わせて探索基準ブロックもブロックサイズは $b/2$ にしたものを用い、輪郭位置補正処理を行うようにする。そして、これにより、演算処理を軽減するようにした。

#### 【 0 0 7 6 】

図1は、本発明の一実施形態を示すフローチャートであり、このフローチャートを参照して本発明を説明する。本発明では、第1段階としてシェイプデータ（アルファマップ）および原画像および探索基準ブロックをサイズ縮小して輪郭位置補正処理を行うようにして、これにより、暫定的なシェイプデータ（アルファマップ）を抽出対象物体の正しい輪郭の形状に近付けておき（粗調整段階）、次に、粗調整された暫定的シェイプデータ（アルファマップ）を抽出対象物体の正しい輪郭の形状にフィットさせるために、この粗調整されたシェイプデータ（アルファマップ）および原画像については本来のサイズで使用し、探索基準ブロックについては粗調整段階で使用した小さいサイズのものを使用して輪郭位置補正処理を行うようにする（本調整段階）ことで、演算量を大幅に低減した輪郭抽出処理を実施可能にする。

#### 【 0 0 7 7 】

本発明の要部を説明する。

最初の段階でのシェイプデータ（アルファマップ）、すなわち、暫定的なシェイプデータ（暫定的なアルファマップ）が得られたならば、まず、粗調整段階では、前述の「処理2-2」における処理である「シェイプデータ（アルファマップ）の輪郭（抽出対象物体の概略形状2）に沿って探索基準ブロックB1, B2, ..., B<sub>n-1</sub>, B<sub>n</sub>を配置する」と云う処理に先立ち、探索基準ブロックのブロックサイズを定める。

## 【0078】

【ステップS11】：既存技術と同様、探索基準ブロックのブロックサイズbを初期値はAとする。

## 【0079】

【ステップS12】：最初の段階でのシェイプデータ（アルファマップ）である暫定的なシェイプデータ（暫定的なアルファマップ）を縦横1/2に縮小する。この縮小は元となる暫定的なシェイプデータ（暫定的なアルファマップ）について1画素とびの間引きを行うことにより、あるいは、サンプリング点近傍4画素の多数決フィルタなどを用いてフィルタ処理することで得られる。

## 【0080】

【ステップS13】：フレーム画像（原画像）を縦横1/2に縮小処理し、また、探索基準ブロックのブロックサイズを縦横1/2に縮小処理する。

そして、この縮小処理済みのフレーム画像（原画像）、探索基準ブロック、シェイプデータ（アルファマップ）を用いて前述の【処理2-2】以降の輪郭位置補正処理を行う。

## 【0081】

【ステップS14】：次に、上述の輪郭位置補正処理で用いた探索基準ブロックのブロックサイズをさらに縦横1/2に縮小処理する。

【ステップS15】：そして、S14において縮小処理された探索基準ブロックのブロックサイズbがYよりも小さいか否かを調べる。Yは予め設定した所望の値であり、 $Y < A$ の関係にある。bとYとの大小比較の結果、bがYよりも小さい場合にはステップS16の処理に進む。そうでない場合にはステップS13の処理に戻る。これは粗調整での輪郭位置補正処理に用いる探索基準ブロックのブロックサイズが縮小され過ぎて探索基準ブロック内の画像が少なくなり過ぎるのを避けるための処置である。

## 【0082】

このようにしてブロックサイズが所定のサイズに到達するまでこの処理を繰り返すことで、暫定的なシェイプデータ（暫定的なアルファマップ）を抽出対象物体の正しい輪郭の形状に大まかに近付けることができる（粗調整段階）。

## 【 0 0 8 3 】

b と Y との大小比較の結果、b が Y よりも小さい場合には粗調整段階終了であり、本調整段階に処理を移すことになる。本調整段階はステップ S 1 6 以降の処理が該当する。

## 【 0 0 8 4 】

〔ステップ S 1 6〕：本調整段階ではまずはじめに粗調整段階では縮小して使用していたシェイプデータ（アルファマップ）を本来のサイズに戻すことを行う。すなわち、粗調整段階で使用して粗調整されたシェイプデータ（アルファマップ）である粗調整済みシェイプデータ（粗調整済みアルファマップ）のサイズを縦横 2 倍に拡大する。これにより本来の原画像と同じサイズになる。

## 【 0 0 8 5 】

〔ステップ S 1 7〕：本来のサイズの原画像を用い、また、粗調整段階最後の処理で使用したブロックサイズ b の探索基準ブロックと本来のサイズに戻された粗調整済みのシェイプデータ（粗調整済みアルファマップ）とを用いて輪郭位置補正処理を行う。

## 【 0 0 8 6 】

〔ステップ S 1 8〕：次に、探索基準ブロックのブロックサイズ b の値を  $1/2$  にする。

## 【 0 0 8 7 】

〔ステップ S 1 9〕：探索基準ブロックのブロックサイズ b が Z よりも小さいか否かを調べる。Z は予め設定した所望の値であり、 $Z < Y$  の関係にある。b と Z との大小比較の結果、b が Z よりも小さい場合には処理を終了し、b が Z よりも大きい場合にはステップ S 1 7 の処理に戻る。

## 【 0 0 8 8 】

これは本調整における輪郭位置補正処理に用いる探索基準ブロックのブロックサイズが縮小され過ぎて探索基準ブロック内の画像が少なくなり過ぎるのを避けるための処置である。探索基準ブロックのサイズが次第に小さくなることで、細かい凹凸までフィットさせることが可能になる。

## 【 0 0 8 9 】

このようにして探索基準ブロックのサイズが小さい所定サイズに到達するまでこの処理を繰り返すことで、シェイプデータ（アルファマップ）を抽出対象物体の正しい輪郭の形状に細かいところまで近付けることができる。

【0090】

＜輪郭位置補正処理フロー＞

ここで、ステップS13とステップS17での処理である輪郭位置補正処理の詳しいフローチャートを図2に示す。なお、S13では原画像の1/2の画像サイズ、S17では原画像と同じ画像サイズでの処理である。

【0091】

〔ステップS21〕：暫定的なシェイプデータ（暫定的なアルファマップ）の輪郭部分に図5のようにブロックを設定する。ここで、S13では、探索基準ブロックの一辺のサイズが $b/2$ なるブロックサイズ、S17では探索基準ブロックの一辺のサイズが $b$ なるブロックサイズとする。

【0092】

〔ステップS22〕：画像データを用いて探索基準ブロックの相似ブロックを求める。

【0093】

〔ステップS23〕：ステップS21で設定した全ての探索基準ブロックについて相似ブロックを求め終わった場合はステップS24の処理に進む。そうでない場合には他の探索基準ブロックの相似ブロックを求めるためにステップS22の処理に戻る。

【0094】

〔ステップS24〕：探索基準ブロックのシェイプデータを、相似ブロックのシェイプデータを縮小したもので置き換える。

【0095】

〔ステップS25〕：全ての探索基準ブロックについて置き換えが終わった場合はステップS26の処理に進む。そうでない場合には、他の探索基準ブロックの置き換えを行うためにステップS24の処理に戻る。

【0096】

【ステップ S 2 6】：置き換えの回数が所定の回数に達した場合は終了し、そうでない場合は、ステップ S 2 4 の処理に戻る。

## 【 0 0 9 7】

本発明において、この輪郭位置補正処理は、例えばフレーム画像（原画像）のサイズが  $320 \times 240$  画素、 $A = 32$ 、 $Y = 16$ 、 $Z = 4$  の場合、下表のようになる。なお、 $A$ 、 $Y$ 、 $Z$  のこのような値は、適宜に設定した値である。 $A$  は本来のサイズであり、 $Y$  は粗調整段階で粗調整としての輪郭位置補正処理をどこで打ち切るかその閾値として使用し、また、 $Z$  は本調整段階での輪郭位置補正処理をどこで打ち切るかその閾値として使用する。

## 【 0 0 9 8】

表：

【回】	【b】	【画像サイズ】	【ブロックサイズ】
i 回目	32	$160 \times 120$	16
ii 回目	16	$160 \times 120$	8
iii 回目	8	$320 \times 240$	8
iv 回目	4	$320 \times 240$	4

すなわち、探索基準ブロックの本来のブロックサイズ  $b = A$  が “32” であったとして、粗調整段階では “ $320 \times 240$  画素” 構成のフレーム画像（原画像）を第 1 回目の処理（表における i 回目）では  $1/2$  のサイズにした “ $160 \times 120$  画素” 構成で使用し、そのときの探索基準ブロックは縦および横がそれぞれ 16 画素のブロックサイズとなっている。そして、この第 1 回目の処理が終了した段階では  $b$  は “32” であるので、これを  $1/2$  して “16” を得、これを  $Y$  の値と大小比較する。この “16” なる値は、“16” に設定した  $Y$  の値より小さくはないので、粗調整段階での第 2 回目（表における ii 回目）の処理に移る。粗調整段階での第 2 回目の処理では第 1 回目の処理で使用したものと同様に “ $320 \times 240$  画素” 構成のフレーム画像（原画像）の  $1/2$  のサイズである

“160×120画素”構成の縮小版フレーム画像（原画像）使用し、そのときの探索基準ブロックは第1回目の半分のサイズである縦および横がそれぞれ8画素のブロックサイズとなっている。

#### 【0099】

そして、この第2回目の処理が終了した段階ではbは“16”であるので、これを1/2して“8”を得、これをYの値と大小比較する。“8”なる値は、

“16”に設定したYの値より小さいので、粗調整段階は終了し、本調整段階に移る。

#### 【0100】

本調整段階では、bは粗調整段階最終処理での値を踏襲し、また、探索基準ブロックのサイズも粗調整段階最終処理での値を踏襲するが、フレーム画像（原画像）は本来のサイズのものをを用いる。すなわち、この例では“320×240画素”構成のフレーム画像（原画像）を用い、探索基準ブロックのサイズは“8”、bは“8”である。

#### 【0101】

本調整段階では“320×240画素”構成のフルサイズフレーム画像（原画像）を毎回の処理に使用する。本調整段階での第1回目（表におけるiii回目）の処理では探索基準ブロックは縦および横がそれぞれ8画素のブロックサイズとなっている。そして、この第1回目（表におけるiii回目）の処理が終了した段階ではbは“8”であるので、これを1/2して“4”を得、これをZの値と大小比較する。この“4”なる値は、“4”に設定したZの値より小さくはないので、本調整段階での第2回目（表におけるiv回目）の処理に移る。本調整段階での第2回目（表におけるiv回目）の処理でも本調整段階での第1回目（表におけるiii回目）の処理で使用したものと同様に“320×240画素”構成のフレーム画像（原画像）のフルサイズフレーム画像（原画像）使用し、そのときの探索基準ブロックは本調整段階での第1回目の処理での半分のサイズである縦および横がそれぞれ4画素のブロックサイズとなっている。

#### 【0102】

そして、この第2回目（表におけるiv回目）の処理が終了した段階ではbは

“4”であるので、これを $1/2$ して“2”を得、これをZの値と大小比較する。“2”なる値は、“8”に設定したZの値より小さいので、本調整段階も終了条件が整うので終了して処理を完了する。

## 【0103】

このように本発明において、輪郭位置補正処理は、フレーム画像（原画像）のサイズが $320 \times 240$ 画素、 $A=32$ 、 $Y=16$ 、 $Z=4$ なる条件である場合、粗調整段階処理を2回、本調整段階処理を2回、計4回で処理を完了するが、粗調整は演算に使用する画素数が毎回、半減し、本調整段階ではフレーム画像（原画像）およびシェイプデータ（アルファマップ）についてはフルサイズのものに戻すものの、探索基準ブロックは粗調整段階からのサイズを処理毎に $1/2$ ずつにサイズ変更していくので、トータルとしての演算処理の負担は大幅に軽減される。しかも、探索基準ブロックのサイズを粗調整段階から継続して、処理毎に $1/2$ ずつにサイズ縮小できるのは、粗調整段階でシェイプデータ（アルファマップ）が輪郭抽出対象物体の輪郭にほぼ近づけることができているからである。

## 【0104】

従って、自己相似写像を用いる手法を採用して輪郭抽出する場合に、はじめに与えた抽出対象物体の概略形状を、少ない演算処理によって、輪郭抽出対象物体の輪郭に一致させることができるようになり、暫定的に与えたシェイプデータの輪郭を、輪郭抽出対象物体の輪郭に高速で一致させることができる輪郭抽出方法が得られる。

## 【0105】

なお、上記例において、探索基準ブロックのブロックサイズbは必ずしも $1/2$ ずつ小さくする必要はない。例えば、“32”、“8”、“4”の如きとしてもよく、このようにすれば、処理量をさらに削減できるようになる。

## 【0106】

以上は、シェイプデータ（アルファマップ）の輪郭位置調整処理を粗調整段階、本調整段階の2段階で実施すると共に、粗調整段階では、原画像も探索基準ブロックもまた、シェイプデータ（アルファマップ）も縮小したものを扱い、また、縮小するブロックサイズが所定の条件に達するまで前回の輪郭位置調整処理で



の縮小よりもさらに縮小させて輪郭位置調整処理を実施し、条件に到達したならば次に本調整段階に入り、この本調整段階では原画像もまた、シェイプデータ（アルファマップ）も本来のサイズのものを使用し、探索基準ブロックは粗調整段階の最終時点で使用したサイズのものを使用し、以後、ブロックサイズが所定の条件に達するまで、探索基準ブロックについては毎回、前回より縮小したものをを用いて輪郭位置調整処理を実施し、これによって所望に与えたシェイプデータ（アルファマップ）の形状を抽出対象の輪郭形状に合わせるようにしたものである。

## 【0107】

この輪郭位置調整処理において、相似ブロックを探索するに際して、相似候補を探索基準ブロックの上下左右方向に暫時、拡大して該当のものを探していくようにしているが、常に上下左右方向に拡大するのではなく、探索基準ブロック内のシェイプデータの輪郭の存在する方向に従ってその方向対応に拡大していくようにすればより合理的である。その例を次に第2の実施形態として説明する。

## 【0108】

## （第2の実施形態）

相似ブロックの探索範囲を探索基準ブロック内のシェイプデータの輪郭の方向によって切り替える実施形態を説明する。

## 【0109】

図7を用いて説明したように、従来においては、相似ブロックの探索範囲は、探索基準ブロックとの相対的な位置によって決まり、画面の場所やシェイプデータや画像データによっては制御されない。

## 【0110】

しかし、例えば、図3に示した探索基準ブロックB1のように、シェイプデータの輪郭がその内部を横切っている場合には、左右方向を省略し、上下方向へW画素分、動かした範囲Bs1'で探索するだけでも、抽出性能はほとんど劣化しない。

なぜなら、相似ブロックBs1のような場合には、シェイプデータの輪郭は置き換えのときに上下方向に移動して初めて補正の効果が得られるものであり、従

って、この例の場合、上下方向に移動させるのは、そのために上下方向にずれた相似ブロックを選べるようにしたいがためである。

#### 【0 1 1 1】

背景や物体の細かい図柄の影響で、左右にずらした場合の誤差が最小となる場合もあるが、それよりも誤差は多少大きくても、シェイプデータの輪郭が上下方向に移動する方が正しい輪郭に近づく。

#### 【0 1 1 2】

輪郭抽出対象物体（正しい物体領域）1の輪郭1 aは、もちろん未知であり、暫定的に与えたシェイプデータの輪郭（抽出対象物体の概略形状2）をどちらに動かせば輪郭抽出対象物体（正しい物体領域）1の輪郭1 aに近づくのかも、相似ブロック探索時には当然分からないのであるが、シェイプデータの輪郭（抽出対象物体の概略形状2）の方向と輪郭抽出対象物体（正しい物体領域）1の輪郭1 aの方向は経験的にみて、ほぼ合っていることがほとんどなので、シェイプデータの輪郭（抽出対象物体の概略形状2）の方向と垂直な方向に相似候補ブロックを動かすのが最も合理的である。

#### 【0 1 1 3】

例えば、シェイプデータのブロックの左上、右上、左下、右下の4画素の値を比較し、左上と右上が等しく、左下と右下が等しい場合には横方向の輪郭と判定し、相似候補ブロックは上下方向のみの範囲を動かしていき、相似ブロックを探索していく。

同様に、左上と左下が等しく、右上と右下が等しい場合には縦方向の輪郭と判定し、左右方向に相似候補ブロックを変化させて相似ブロックを探索する。

#### 【0 1 1 4】

また、そのいずれでもない場合は、従来方法と同様に、全方向で探索する。

このようにすることで抽出精度を劣化させずに相似ブロックの探索の処理量を削減できる。

#### 【0 1 1 5】

ここで、相似ブロックの探索時には、探索範囲内で誤差最小の相似候補ブロックを以てそこでの相似ブロックに決定するのではなく、順次相似候補ブロックを

変えて誤差評価をする際に、予め定める許容誤差よりも誤差が小さくなった相似候補ブロックが見つかった時点で探索を打ち切り、その相似候補ブロックを相似ブロックとして決定すると良い。

## 【0116】

このようにすることにより、抽出精度を劣化させずにさらに処理量を削減できるようになる。

## 【0117】

また、ブロックの画面内での位置によって、探索範囲を切り替えるのが有効な場合もある。例えば、特願平11-186537号「画像輪郭抽出装置」で提案したように、抽出対象が頭部を含めた自己の顔画像部分であったとして、この場合、人間の頭部輪郭線像を予め用意しておき、画面にこの頭部輪郭線像を表示させると共に、画面に表示されたこの頭部輪郭線像としての枠線内に自己の頭部を合わせるように位置決めして撮像し、その枠線をシェイプデータの初期状態（暫定的なシェイプデータ（暫定的なアルファマップ））として用い、画像中から頭部を輪郭に沿って抽出する技術があるが、この技術を適用する場合において、ユーザが常にあごの位置を枠線の下限に合わせるとすると、あごの部分の輪郭のずれは他の部分よりも小さくなる。

## 【0118】

このような時には、枠線の下の部分では、大きいブロックサイズでの処理を省略したほうが、誤抽出の確率が低くなる。あるいは、探索範囲を狭くしたほうが、やはり誤抽出の確率が低くなる。大きいブロックでの処理の省略や探索範囲を狭くすることは処理量の削減にもなる。また、上のように初期シェイプが既知の場合には、探索基準ブロックの配置も一意に決まるので、その配置を記憶しておき、探索基準ブロックを配置するステップS21では、それを読み出すだけにすれば、シェイプデータの輪郭を検出する処理を省略できる。

## 【0119】

また、図5のように、数珠つなぎになった探索基準ブロック $B_1, \dots, B_n$ の全てについて相似ブロックを求めるようなことはせずに、間欠的に、例えば、一つ置きに求めるようにし、間のブロックの相似ブロックはその両端のブロックの相

似ブロックの中間位置にあるものとして処理することで、探索の処理を大幅に省略できる。

#### 【0120】

また、各探索基準ブロックが互いが重ならないように配置する方法もある。このようにすると、探索基準ブロックの総数が減ることから、当然のことながら処理量が減ることになる。

#### 【0121】

また、相似候補ブロックを設定するたびに縮小処理を施すのではなく、予め画面全体を縮小したものを用意しておき、相似候補ブロックを設定したときに、それに対応する部分を当該縮小画像から取り出して、探索基準ブロック内の画像との誤差比較をする方法もある。このようにすることによっても、全体の処理量を削減できるようになる。

#### 【0122】

このとき、例えば、探索基準ブロックと相似ブロックのサイズが1対2で、相似候補ブロックを1画素ずつずらしながら設定する場合、図4に円で示した如くA、B、C、Dなる原画像の画素位置に対して、Aの画素位置でサンプリングした画素を用いて縦横半分のサイズの縮小画面を作成する。また、サンプリング点がB、C、D点の画素位置にずれた、つまり、位相がずれた場合のそれぞれの画像の縮小画像を作る。これにより、合計4枚の縮小画面が作られたことになる。このとき、相似候補ブロックの位置を例えば縦座標、横座標とも偶数値だけにし、つまり、探索範囲内で2画素ずつずらしながら探索する場合には、A、B、C、Dのうち、一つだけを予め作れば十分であり、この場合、他の3枚を作る処理を省略できる。

#### 【0123】

1画素ずらしを2画素ずらしにするとそれだけ相似探索のための相似候補ブロックの数が減り、探索処理は少なくなるが、縮小画面を用いる方法では、さらに縮小画面を作る処理も大幅に削減できる。

#### 【0124】

また、初期シェイプデータが小さい場合、縮小画面は画面全体を作る必要はな

い。予め、相似ブロックの探索範囲には入らない部分がある場合には、その部分の縮小画面を作る処理を省いて良い。

#### 【 0 1 2 5 】

なお、以上においては、相似ブロック探索時の縮小画面の生成について述べたが、見つけた相似ブロックの画像をシェイプデータ（アルファマップ）上に置き換え変換する時も同様に、最初に縮小したシェイプ画面を作っておき、各探索基準ブロックのシェイプデータをその縮小したシェイプ画面から抜き取ってきて置き換えるようにできる。

このようにすると、置き換えのたびに縮小処理を省略できるようになり、演算処理の一層の負担低減を図ることができるようになる。

#### 【 0 1 2 6 】

以上、第 2 の実施形態によれば、相似ブロックを探索するにあたり、その探索範囲を探索基準ブロック内のシェイプデータ（アルファマップ）の輪郭方向と垂直の方向に制限することで、暫定的に与えたシェイプデータ（アルファマップ）の輪郭を輪郭抽出対象物体の輪郭に近付ける処理を合理的に行うことができ、しかも、演算処理量を大幅に軽減できる。また、シェイプデータ（アルファマップ）の輪郭上に配置する探索基準ブロックは互いが重ならないようにすることで、演算処理量を大幅に軽減できる。また、相似候補ブロックを設定するたびに縮小処理を施すのではなく、予め画面全体を縮小したものを用意しておき、相似候補ブロックを設定したときに、それに対応する部分を当該縮小画像から取り出して、探索基準ブロック内の画像との誤差比較をするようにすることにより、全体の演算処理量を削減できるようになる。

#### 【 0 1 2 7 】

ところで、画像から被写体の領域を抽出する場合、利用者がマニュアル操作によって物体のおよその形状を描くか、あるいは、利用者が画面を見ながら、そこに表示される輪郭線に自分の頭部を合わせることで、物体のおよそのシェイプデータを得て、次に、そのシェイプデータの輪郭を画像データ内の物体の輪郭の位置に補正する方法が取られ、これらにより、ほとんどの状況において正確な抽出が行われる。この場合、画面中央の楕円の領域を表すシェイプデータが生成され

、利用者が画面に表示される楕円の輪郭線に自分の頭部を合わせることで、楕円のシェイプデータは次のステップで頭部の領域を抽出するに足る程度に頭部を近似している。楕円のシェイプデータを初期シェイプとして、そのシェイプデータの輪郭の位置が画像データ内の頭部の輪郭の位置に補正される。

## 【 0 1 2 8 】

しかしながら、初期シェイプデータを得るために利用者が自分で画面内の所定の位置に頭部を固定するのは不便である。背景が比較的平坦であったり、背景に動きがない場合など、物体の抽出が比較的容易な状況では、自動的に初期シェイプデータが得られることが望まれる。

## 【 0 1 2 9 】

そこで、この要望に沿った実施形態を以下に説明する。

## 【 0 1 3 0 】

## (第 3 の実施形態)

図 1 0 に第 3 の実施形態の画像からの物体抽出方法を説明するフローチャートを示す。本実施形態では分離度という統計的指標を用いる。分離度  $S$  は、次式により表される(福井「領域間の分離度に基づく物体輪郭抽出」、電子情報通信学会論文誌、D-II、V o 1 . J 8 0 - D - II、No. 6、pp. 1 4 0 6 - 1 4 1 4、1 9 9 7)。即ち、

$$S = X / Y$$

$$X = N_b * (A_b - A) * (A_b - A) + N_f * (A_f - A) * (A_f - A)$$

$N_b$  : 背景領域の画素数

$N_f$  : 物体領域の画素数

$A$  : 全画素値の平均値

$A_b$  : 背景領域の画素値の平均値

$A_f$  : 物体領域の画素値の平均値

$Y$  : 各画素値と  $A$  の差分の二乗の全画素の和

蒸気式から明らかなように物体領域と背景領域の平均値の差が大きいほど、1 に近づき、それらの平均値の差が小さいほど 0 に近づく。通常、物体と背景では画素値の平均値などの統計的性質は互いに異なるので、分離度が大きいほどより

正しい物体領域であると推定できる。

#### 【0131】

そこで、予めシェイプデータの候補領域を図11に示したように、シェイプ1～10の10個を用意し、これらの各々に対して分離度を求め、その値が最も大きいシェイプデータを探索し、それを初期シェイプデータとするのが本実施形態である。図10のフローチャートに従ってこの処理を説明する。

#### 【0132】

先ず、分離度の最大値を保持する変数Mにゼロを代入する(S31)。数字の順に初期シェイプデータの候補を逐次設定する(S32)。そのシェイプデータを用いた時の現入力画像データにおける分離度Sを求める(S33)。SがMよりも大きい場合にはステップS35に進み、そうでない場合には、ステップS36に進む。

#### 【0133】

ステップS35では、今、分離度を求めたシェイプデータを初期シェイプデータとして記憶する。それ以前に初期シェイプデータとして記憶していたものがある場合は、以前のものを破棄する。また、 $M=S$ とする。

#### 【0134】

全ての初期シェイプデータの候補に対する処理が終わったかが判断される(S36)。処理が終わっていれば、ステップS17に進み、そうでない場合は、ステップS32に戻る。ステップS37では、初期シェイプデータと現入力画像データを用いて、シェイプデータの輪郭線を画像データ内の物体の輪郭線に一致させる。

#### 【0135】

物体の動きが比較的ゆるやかであることが分かっている場合には、初期シェイプデータを常に画面全体で探索するのではなく、例えば毎フレームで選択された初期シェイプデータ候補を記憶しておき、前のフレームで選択された初期シェイプデータの近くだけで探索するようにすれば、処理量を削減できる。例えば、前のフレームでは図12のシェイプ11が選択されたとすると、図示したように、その上下左右の4つとそれ自身の5個で探索すれば、10個で探索する場合より

も探索の処理量を半分に減らせる。あるいは、シェイプ11の近くに互いに少しずつずらしながら10個のシェイプデータ候補を設定すれば、同じ処理量でより高い精度の初期シェイプデータが得られる。このように、前フレームのシェイプを中心に初期シェイプデータの探索範囲を設定して人物の頭部を抽出する場合、探索範囲は、画面の上下方向よりも左右方向に広くする。なぜなら、頭部は上下に動くよりも左右に揺れる確率が高いからである。

## 【0136】

また、分離度を求める際の領域はシェイプデータと全く同じである必要はない。楕円よりも長方形の方が、処理を節約できる場合には、図13(a)に示したような長方形の領域を用いて分離度を求め、その分離度が最大になった長方形の角をおとして図13(b)のような八角形を初期シェイプデータとする方法もある。角をおとすのは、人物の頭部の形に多少でも近づけるためである。

## 【0137】

また、人物の頭部はおおまかに上方の髪と中央以下の顔の2つの領域に分かれる。そこで、図13(c)のように長方形をさらに2つの領域に分け、分離度Sを以下の式により求める。

## 【0138】

$$S=X/Y$$

$$X=N_b * (A_b - A) * (A_b - A) + N_{f1} * (A_{f1} - A) * (A_{f1} - A) + N_{f2} * (A_{f2} - A) * (A_{f2} - A)$$

$N_b$  : 背景領域の画素数

$N_{f1}$  : 第1の物体領域の画素数

$N_{f2}$  : 第2の物体領域の画素数

$A$  : 全画素値の平均値

$A_b$  : 背景領域の画素値の平均値

$A_{f1}$  : 第1の物体領域の画素値の平均値

$A_{f2}$  : 第2の物体領域の画素値の平均値

$Y$  : 各画素値とAの差分の二乗の全画素の和

これにより、より正確に人物の頭部を検出できることがある。



## 【 0 1 3 9 】

ステップ S 3 7 の輪郭位置補正処理は、図 1 4 のフローチャートに従って行われる。即ち、先ずシェイプデータの輪郭部分にブロックが設定される（S 4 1）。次に、画像データを用いてブロックの相似ブロックを求める（S 4 2）。ステップ 4 1 で設定した全てのブロックについて相似ブロックを求め終わったかが判断され（S 4 3）、終わった場合は、処理はステップ S 4 4 に進む。そうでない場合には他のブロックの相似ブロックを求めるためにステップ S 4 2 に戻る。

## 【 0 1 4 0 】

ステップ S 4 4 では、ブロックのシェイプデータを相似ブロックのシェイプデータを縮小したもので置き換える。全てのブロックについて置き換えが終わったかが判断され（S 4 5）、終わっていれば、処理はステップ S 4 6 に進む。そうでない場合には、他のブロックの置き換えを行うためにステップ S 4 4 に戻る。ステップ S 4 6 での判定で、置き換えの回数が所定の回数に達した場合は処理は、終了する。そうでない場合は、処理はステップ S 4 4 に戻る。

## 【 0 1 4 1 】

背景に初期シェイプデータ候補のいずれかに合う物体があると、これにおける分離度の方が、頭部における分離度よりも高くなることがある。しかし、背景が比較的平坦な場合には、本実施形態により頭部が画面のどこにあっても良好な初期シェイプが得られる。

## 【 0 1 4 2 】

## （第 4 の実施形態）

次に、図 1 5 および図 1 6 を参照して、第 4 の実施形態を説明する。本実施形態では、抽出を始める前に背景画像を取得しておき、その後は、入力画像と背景画像の差が一定値以上となる領域を物体領域と判定することで初期シェイプデータを生成する方法である。

## 【 0 1 4 3 】

先ず、抽出すべきオブジェクトが撮影範囲に入っていない画像（図 1 6（a））を背景画像として利用者が確認しながら取得する（S 5 1）。次に、図 1 6（b）に示すようにオブジェクト（人物）が画面に入ると、このとき、現入力フレ

ームと背景画像の差分を求め、差分の絶対値が一定値以上となる画素を物体画素、そうでない画素を背景画素として初期シェイプデータ（図 1 6（c））を生成する（S 5 2）。

#### 【 0 1 4 4 】

ただし、単純に画素単位でこの処理を行うと領域が粒状に分割される場合が多い。そこで、背景画像をブロックに分割し、それぞれのブロックにおいて画素値のブロック平均値を求めておく。入力画像でも同じくブロック平均値を求め、入力画像と背景画像のブロック平均値の差の絶対値が所定値よりも大きいブロックを物体領域とし、そうでないブロックを背景領域として初期シェイプデータを生成する。このようにブロック単位にすると、輪郭に段差があるシェイプデータしか得られないが、ステップ S 5 3 において輪郭補正を行うので問題はない。さらに、先に述べた画素単位で求めた差分の絶対値のブロック内での和を求め、このブロック内での和が所定値よりも大きい場合に、そのブロックも物体領域に加える。これにより、本来、背景と図柄が異なる物体領域が、ブロック平均値をとったために背景のブロック平均値に近くなってしまい、物体と判定されないという誤検出を回避できる。

#### 【 0 1 4 5 】

ステップ S 5 3 では、初期シェイプデータと現入力画像データを用いて、シェイプデータの輪郭線を画像データ内の物体の輪郭線に一致させ、処理を次のフレームに進める（S 5 4）。背景が静止している場合には、本実施形態によって、どのような形状の物体でも、あるいは、複数の物体でも、それらが画面内のどこにあっても高い精度で抽出できる。

#### 【 0 1 4 6 】

##### （第 5 の実施形態）

次に、基準フレームとの動き補償によって初期シェイプデータを生成する第 5 の実施形態を図 1 7 を用いて説明する。

#### 【 0 1 4 7 】

まず、固定の初期シェイプデータを用いる方法で、利用者が頭部を画面表示に合わせながら頭部の輪郭が抽出される。この輪郭抽出は、正しく抽出されている

ことを利用者が確認しながら画像取得操作を行う。そのときの画像と最終的なシェイプデータをそれぞれ基準画像データ、基準シェイプデータとする（S 6 1）。次に、図 1 8 に示すような基準画像データから現入力フレームの画像データへの動き検出を行う。その動き検出結果を用いて、基準シェイプデータに対して動き補償を行うことで、現入力フレームの初期シェイプデータを生成する（S 6 2）。例えば、基準フレームの物体領域の画像データを切り出し、それを現入力フレーム内で、平行移動しながらマッチングをとる。マッチング誤差が最小となった部分に基準シェイプデータの物体領域を平行移動させて初期シェイプデータを生成する。この後、初期シェイプデータと現入力画像データを用いて、シェイプデータの輪郭線を画像データ内の物体の輪郭線に一致させる、即ち輪郭位置の補正処理が行われる（S 6 3）。次いで、処理は次のフレームに進める（S 6 4）。

#### 【 0 1 4 8 】

本実施形態によって、初めは利用者が画面を確かめながら自分で頭部の位置の調整をしなければならないが、基準画像を取得した後は、動き補償によって、物体の同じ面がカメラに向いているかぎり、画面内のどこにあっても抽出されるようになる。動き検出は平行移動の 2 パラメータに限らず、回転を含めた 3 パラメータ、さらに拡大/縮小を含めた 4 パラメータ、アフィン変換の 6 パラメータなど、パラメータ数を増やせば処理量は多くなるが、精度は向上する。

#### 【 0 1 4 9 】

物体が最初の位置からあまり動かないことが予め分かっている場合は、基準フレームは最初のままで変更する必要はない。しかし、物体が最初の位置から徐々に動いていく場合には、基準フレームは適宜更新し、例えば、フレーム毎に更新し、常に直前のフレームと動き補償を行うようにすれば、物体が時間を経て大きく動くときにも正しく抽出できる。

#### 【 0 1 5 0 】

次に、複数の物体を区別しながら同時に抽出する実施形態を説明する。図 1 9 のようにあるフレームで物体 A と物体 B が離れて抽出されたとして、以降のフレームでもこれらを区別しながら追跡と抽出を行うとする。この場合、第 3 実施形

態や第4実施形態に従って物体をそれぞれ抽出し、フレーム間で重なる画素数が多いもの同士を同じ物体だと判定するのが簡便に実現できる方法である。しかし、これらの方法では、図19(a)の状態だったものが、19(b)、(c)、(d)というように変化し、物体が途中で一部重なると、図19(b)や(c)のときに物体領域が融合してしまい、その結果、図19(d)のときに、どちらが物体Aでどちらが物体Bかが分からなくなる。このような場合に対応するためには、第5実施形態を2つの物体に対してそれぞれ用い、動き補償を行ったもの同士を同じ物体だと区別する。

#### 【0151】

動き補償によって初期シェイプデータが図19(b)の様に求まったときに、特願平11-001891号や特願平11-301415号に記載された自己相似写像法を用いて輪郭位置の補正を行う例を図20と図21を用いて説明する。ここで、シェイプデータにおいては、物体Aの領域とされた画素には画素値1が代入され、物体Bの画素には画素値2が代入され、物体Aでも物体Bでもない画素には画素値0が代入されているとする。このようなシェイプデータの輪郭、すなわち、0と1の境界、0と2の境界、1と2の境界を画像データにおける物体の輪郭、すなわち、背景と物体Aの境界、背景と物体Bの境界、物体Aと物体Bの境界にそれぞれ補正するのが目的である。先願ではシェイプデータ(アルファマップ、アルファデータ、アルファマスクと同じ)は、2値の画像であったが、ここでは、{0、1、2}の3値の画像であることに注意する。

#### 【0152】

まず、図21に示したようにシェイプデータの輪郭に沿ってブロックを配置する。そのために、シェイプデータをラスタスキャンし、隣接する画素と画素値が異なり、かつ、それまでに設定したブロックに含まれない画素を中心にして逐次、ブロックを設定する。この方法だと、ブロックが互いに重なり合いながら数珠つなぎに配置される。あるいは、特願平11-001891号に示したようにブロックが重ならないように配置する方法もある。次に、画像データを用いてブロックごとに、相似ブロックを探索する。

#### 【0153】

最後に、ブロックのシェイプデータを相似ブロックのシェイプデータを縮小したもので置き換える。例えば、図 20 に示したように、ブロックが  $2 \times 2$  画素で、相似ブロックが  $4 \times 4$  画素の場合、画素 12 に対する相似ブロック内でのサンプリングポイントは点 13 になる。そこで、点 13 の周囲 4 画素の画素値を調べ、最も数が多かった画素値 (0、1、2 のいずれか) をサンプリング値として、この値で画素 12 を置き換える。画素 14 に対しても点 15 の周囲 4 画素を用いて同様に置き換える。このようにして全ブロックの画素値の置き換えを複数回繰り返すことで、シェイプデータの輪郭は画像データ内の物体の輪郭に近づき、それと一致した状態で収束する。

#### 【0154】

この方法によれば、物体を区別しながら追跡して、なおかつ輪郭に沿った抽出を行える。なお物体の数が 3 つ以上の場合はそれに応じて画素値の種類すなわちラベルを増やす。ラベルが増えても、サンプリング値を多数決で決めるのは同じである。

#### 【0155】

また、ブロックが重なる部分では、一つの画素に複数のサンプリング点に対応する。その場合には、上述したブロック毎の置き換えにおいて最後に置き換えた値を有効とする。あるいは、複数のサンプリング点の周囲の画素、例えば 3 個のサンプリング点の周囲の合計 12 個の画素を用いた多数決で置き換える値を決める。

#### 【0156】

本実施形態で用いた、3 つ以上のラベルを用いる輪郭抽出補正法は、画像のセグメンテーションに用いることもできる。その際、ブロックの配置は図 22 (a) に示したように、画面全体を分割したものとする。次に、この画像データを用いて各ブロックに対して相似ブロックを求めるのは図 14 のステップ S42 等と同じである。そして、初期シェイプデータとしては、例えばランダムに各画素にラベルを配置したものを用意する。ラベルの数は予め決めておく。あるいは、画面を正方形のブロックに分割してブロック毎にラベルをつける。このブロック分割は図 14 と同じ方法であっても異なっても良い。あるいは、画像データの画素

値を量子化し、各量子レベルにラベルを割り当てる。

【 0 1 5 7 】

このようにして作成した初期シェイプデータに対して、相似ブロックからブロックへの置き換えを繰り返すとあるシェイプデータに収束する。セグメンテーション画像は、収束したシェイプデータのラベルごとに色分けした画像として図 2 2 ( b ) のように得られる。第 5 の実施形態において基準フレームを設定する際に、このセグメンテーション画像を表示し、利用者が物体に属する領域を一つずつ選択することでその基準フレームの物体領域を設定してもよい。

【 0 1 5 8 】

( 第 6 の実施形態 )

次に、図 2 3 、図 2 4 および図 2 5 を参照して本発明の物体抽出方法を用いた画像伝送システムを第 6 の実施形態として説明する。

【 0 1 5 9 】

図 2 3 において、具体的には、例えば携帯電話やパソコン、ゲーム機である画像送受信端末 X は、端末付属のカメラ 1 6 、画像切り出し器 1 8 、符号器 2 0 、復号器 3 6 および表示器 3 8 を備えている。他の画像送受信端末 Y 、 Z も端末 X と同様に構成されている。

【 0 1 6 0 】

これら画像送受信端末 X 、 Y 、 Z に接続される画像配信センター A は、端末 X 、 Y 、 Z にそれぞれ接続される復号器 2 4 、 2 5 、 2 6 、これら復号器に接続される合成器 3 0 、背景メモリ 3 1 および符号器 3 4 を備えている。

【 0 1 6 1 】

このような構成において、端末付属のカメラ 1 6 によって端末 X の利用者の画像が撮影され、その画像データ 1 7 が切り出し器 1 8 に送られる。切り出し器 1 8 では利用者の顔画像が第 3 、 4 、 5 の実施形態などの手法で切り出され、顔画像データ 1 9 が符号器 2 0 に送られる。顔画像データとは、画像データ 1 7 と同じものと、その顔領域を表すシェイプデータ ( アルファマップ、アルファデータ、アルファマスク ) をあわせたものである。符号器 2 0 では、例えば、動画像符号化の国際標準である M P E G - 4 方式によって顔画像データがオブジェクト符

号化され、圧縮データ 2 1 が通信回線を介して画像配信センター A に送られる。

【 0 1 6 2 】

端末 X とは離れたところにある端末 Y、Z から同様にそれぞれの利用者の顔画像が圧縮され、圧縮データ 2 2、2 3 としてそれぞれ、センター A に送られる。センター A では受信した圧縮データ 2 1、2 2、2 3 がそれぞれ、復号器 2 4、2 5、2 6 で顔画像データ 2 7、2 8、2 9 に再生され、画像合成器 3 0 に送られる。

【 0 1 6 3 】

合成器 3 0 には背景メモリ 3 1 から背景画像 3 2 も入力される。合成器 3 0 では図 2 4 に示したように背景画像 3 2 に顔画像データ 2 7、2 8、2 9 が合成され、この合成画像 3 3 が符号器 3 4 に送られる。符号器 3 4 では、合成画像 3 3 が M P E G - 4 方式などで通常の矩形画像として圧縮され、圧縮データ 3 5 が端末 X、Y、Z のそれぞれに送られる。

【 0 1 6 4 】

端末 X では受信した圧縮データ 3 5 は復号器 3 6 に送られる。復号器 3 6 では合成画像が再生され、その合成画像 3 7 は端末付属の表示部 3 8 に送られ、図 2 4 と同じ画像が表示される。端末 Y と Z にも圧縮データ 3 5 がおくられ、同様の合成画像が端末 Y と Z に表示される。

【 0 1 6 5 】

音声も別途伝送すれば、本システムにより、互いの顔を見ながら、サイバー空間を共有しながらの楽しいリアルタイムチャットシステムを実現できる。

【 0 1 6 6 】

本システムの端末と画像配送センターでの手順は図 2 5 に示されている。同図 2 5 では、左側フローが端末での手順を示し、右側のフローは画像配送センターでの手順を示している。

【 0 1 6 7 】

この手順では、まず、画像を撮像する ( S 7 1 ) 。次に、顔画像を切り出す ( S 7 2 ) 。その後、顔画像データを圧縮し、その圧縮データをセンターに送信する ( S 7 3 ) 。

## 【 0 1 6 8 】

センターは、端末から圧縮データを受信し、顔画像データに再生する（S 7 4）。再生した顔画像データを合成して合成画像を作る（S 7 5）。合成画像を圧縮し、その圧縮データを端末に送信する（S 7 6）。

## 【 0 1 6 9 】

端末は、センターから圧縮データを受信し、合成画像に再生する（S 7 7）。この再生合成画像を端末のディスプレイに表示する（S 7 8）。この後、フローはステップ S 7 1 に戻る。

## 【 0 1 7 0 】

なお、画像配信センターでの S 7 4 では、複数の圧縮データを受信することで、複数の顔画像を合成する。

## 【 0 1 7 1 】

## （第 7 の実施形態）

同様の機能を画像配信センターを介さずに実現する第 7 の実施形態に従った別の画像伝送システムを図 2 6 を参照して説明する。このシステムでは、端末 X において、顔画像データ 1 9 を生成し、符号器 2 0 を介して圧縮データ 2 1 を生成する点は、先のシステムと同じであるので説明は省略する。

## 【 0 1 7 2 】

圧縮データ 2 1 は、端末 Y と Z にそれぞれ通信回線を介して送られる。端末 Y と Z から同様に、圧縮データ 2 2、2 3 が送出され、端末 X に送られる。また、端末 Y と Z の間でも相互に圧縮データの送信を行う。端末 X で受信した圧縮データ 2 2、2 3 はそれぞれ、復号器 3 9、4 0 で復号され、顔画像データ 4 1、4 2 が合成器 4 3 に送られる。合成器 4 3 には、端末 X の利用者の顔画像データ 1 9 も入力され、これらが、背景メモリ 4 4 から送らてくる背景画像 4 5 と合成される。合成画像 4 6 は表示部 4 7 に送られて表示される。端末 Y と Z も上記と同様にデータの送受が行われ、合成画像がそれぞれの表示部に表示される。

## 【 0 1 7 3 】

このシステムでは、各端末での処理が多くなり、複数の相手と同時に通信する必要があるが、画像配信センターが不要で、背景画像や顔画像の配置を各端末で



利用者が好きなように決められる利点がある。

【 0 1 7 4 】

図 2 6 のシステムの端末での手順が図 2 7 に示されている。これによると、最初に、画像を撮像する（S 8 1）。次に、物体を抽出する。即ち、顔画像を切り出す（S 8 2）。切り出した画像、例えば自分の顔画像データを圧縮し、その圧縮データを別の端末に送信する（S 8 3）。このとき同時に、別の端末から圧縮データを受信して相手の顔画像データに再生する。自分の顔画像データと相手の顔画像データを合成して合成画像を作る（S 8 4）。この合成画像を表示する（S 8 5）と、フローはステップ S 8 1 に戻る。

【 0 1 7 5 】

図 2 7 の右側には別の端末での手順が示されているが、この手順は左側の手順と同じである。

【 0 1 7 6 】

また、合成画像を生成する際に、顔画像が正しく得られたかどうか、つまり顔が正しく抽出されてきたかどうかを自動的に判定し、顔画像が得られた場合には背景に合成し抽出に失敗して顔画像が得られていない場合には合成しないようにすれば、顔以外の不要なものが誤って合成されることがなくなる。この判定は、例えば、物体領域の肌色の画素数を数え、その肌色の画素数の物体領域全体に対する割合が一定値以上なら顔画像であり、それ以下なら顔画像ではないと判定する。ここで、肌色の判定は例えば、各画素の色が Y、U、V の 3 色で、おのおの 0 ～ 2 5 5 の値で表された画像の場合、 $U = 110$ 、 $V = 160$  を肌色の基準とし、画像データの U と 110 の差の絶対値が所定値よりも小さく、かつ、V と 160 の差の絶対値が所定値よりも小さい画素を肌色と判定する。

【 0 1 7 7 】

顔画像か否かの判定は合成器 3 0 や 4 3 で行うのが一つの方法であるが、符号器において判定を行い、顔画像でない場合には符号化しないようにすれば通信量を削減できる。

【 0 1 7 8 】

表示部 3 8 や 4 7 は端末とは別の例えばテレビを用いても良い。その際、テレ

ビへの画像の送信はブルートゥースなどの無線を用いると配線の煩わしさがなくてよい。また、背景画像 3 2、4 5 は背景メモリ 3 1、4 4 から読み出すのではなく、テレビなどの放送映像を受信して用いてもよい。そうすると、離れた場所にいる人同士が互いの様子を見ながら一緒に同じテレビ番組を楽しむ。

【0 1 7 9】

配送センター A では、合成する画像として、予めタレントなどの顔画像を用意しておき、端末利用者からの求めに応じてそのタレントの顔を利用者の顔と並べて合成して送信するなどのサービスも可能である。背景画像についても同じく、キャラクタの背景を用意しておき、利用者の希望に応じて背景を選択する。

【0 1 8 0】

なお、本発明は上述した実施形態に限定されるものではなく、種々変形して実施可能である。また、実施形態に記載した手法は、コンピュータに読み取り実行させることのできるプログラムとして、磁気ディスク（フロッピーディスク、ハードディスクなど）、光ディスク（CD-ROM、DVD など）、半導体メモリなどの記録媒体に格納して頒布することもできる。

【0 1 8 1】

【発明の効果】

以上述べたように、本発明によれば、暫定的に与えたシェイプデータ（アルファマップ）の輪郭を輪郭抽出対象物体の輪郭に近付ける処理を、演算量を大幅に軽減して実行できる。従って、その分、高速な輪郭抽出が可能となる画像の輪郭抽出方法を提供できる。

【0 1 8 2】

また、本発明によれば、抽出対象の物体が画面内のどこにあっても自動的に初期シェイプデータが得られ、利用者の操作が不要か、非常に少ない操作で物体を抽出できる。

【図面の簡単な説明】

【図 1】

本発明の第 1 の実施形態に従った画像輪郭抽出方法を説明するためのフローチャート。

【図 2】

図 1 におけるステップ S 1 3 とステップ S 1 7 の処理である輪郭位置補正処理の詳しい処理内容を示すフローチャート。

【図 3】

本発明の第 2 の実施形態における相似ブロックの探索範囲の例を説明するための図。

【図 4】

本発明の第 2 の実施形態における縮小画面のサンプリング位置を説明するための図。

【図 5】

自己相似写像を用いた手法による輪郭位置補正処理で使用するシェイプデータ（アルファマップ）上の探索基準ブロックの配置例。

【図 6】

自己相似写像を用いた手法による輪郭位置補正処理で使用する探索基準ブロックと相似ブロックの例。

【図 7】

自己相似写像を用いた手法による輪郭位置補正処理での従来の相似ブロックの探索範囲の例。

【図 8】

自己相似写像を用いた手法による輪郭位置補正処理でのシェイプデータ（アルファマップ）の補正結果を示す図。

【図 9】

自己相似写像を用いた手法による従来の輪郭位置補正処理内容を示すフローチャート。

【図 1 0】

本発明の第 3 の実施形態に従った画像輪郭抽出方法を説明するためのフローチャート。

【図 1 1】

第 3 の実施形態により生成される複数の初期シェイプデータ候補を示す図。

【図 1 2】

他の初期シェイプデータ候補を示す図。

【図 1 3】

(a) 分離度を求める領域、(b) 初期シェイプデータ、(c) 分離度を求める領域を示す図。

【図 1 4】

輪郭位置補正処理を説明するフローチャート。

【図 1 5】

本発明の第 4 の実施形態に従った画像輪郭抽出方法を説明するためのフローチャート。

【図 1 6】

第 4 の実施形態の画像輪郭抽出方法を説明するための図。

【図 1 7】

本発明の第 5 の実施形態に従った画像輪郭抽出方法を説明するためのフローチャート。

【図 1 8】

第 5 の実施形態の画像輪郭抽出方法を説明するための画面図。

【図 1 9】

物体が交差する様子を説明するための図。

【図 2 0】

画素値のサンプリングを説明するための図。

【図 2 1】

複数の物体を識別するブロックの配置を説明する図。

【図 2 2】

ブロック分割した画面およびセグメンテーション画像を示す図

【図 2 3】

第 6 の実施形態に従った画像伝送システムのブロック図。

【図 2 4】

第 6 の実施形態で形成される合成画像を示す図。

【図 2 5】

第 6 の実施形態の伝送システムでの手順を示すフローチャート。

【図 2 6】

第 7 の実施形態の画像伝送システムのブロック図。

【図 2 7】

第 7 の実施形態の伝送システムでの手順を示すフローチャート。

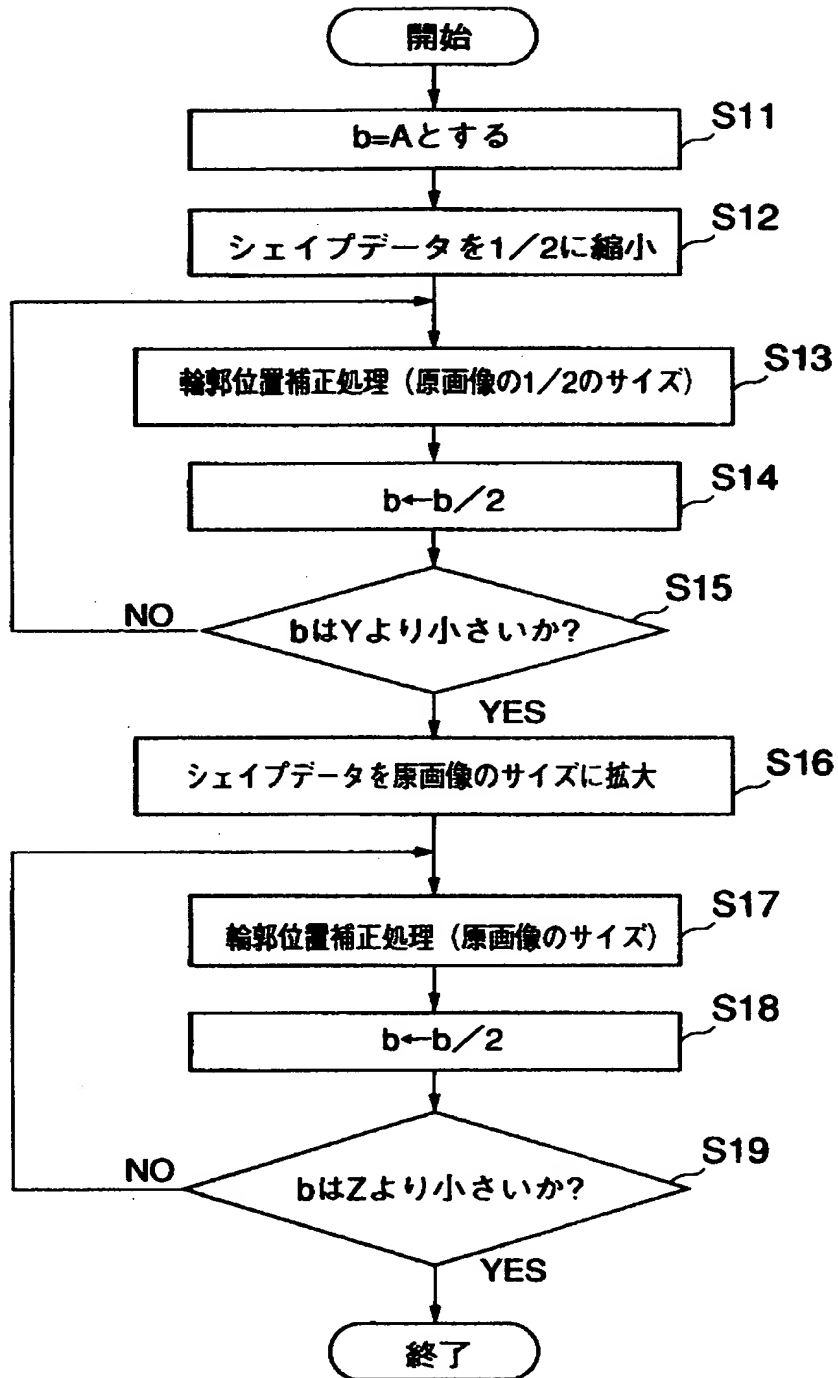
【符号の説明】

- 1 … 輪郭抽出対象物体（正しい物体領域）
- 1 a … 輪郭抽出対象物体の輪郭（正しい物体領域 1 の輪郭）
- 2 … シェイプデータの輪郭（抽出対象物体の概略形状）
- 1 6 … 端末付属のカメラ
- 1 8 … 画像切り出し器
- 2 0、3 4 … 符号器
- 2 4、2 5、2 6、3 6 … 復号器
- 3 0 … 合成器
- 3 1 … 背景メモリ
- 3 8 … 表示器
- B 1、～B n … 探索基準ブロック
- B s 1 ～B s n … 探索基準ブロック B 1 ～B n の相似ブロック
- F s 1 … 探索基準ブロック B 1 の周囲に適宜な大きさの探索領域
- B c … 相似候補ブロック
- B c d … 探索基準ブロック B 1 との誤差評価を行った結果、誤差最小で相似ブロックとして決定されることとなった相似候補ブロック
- X、Y、Z … 画像送受信端末

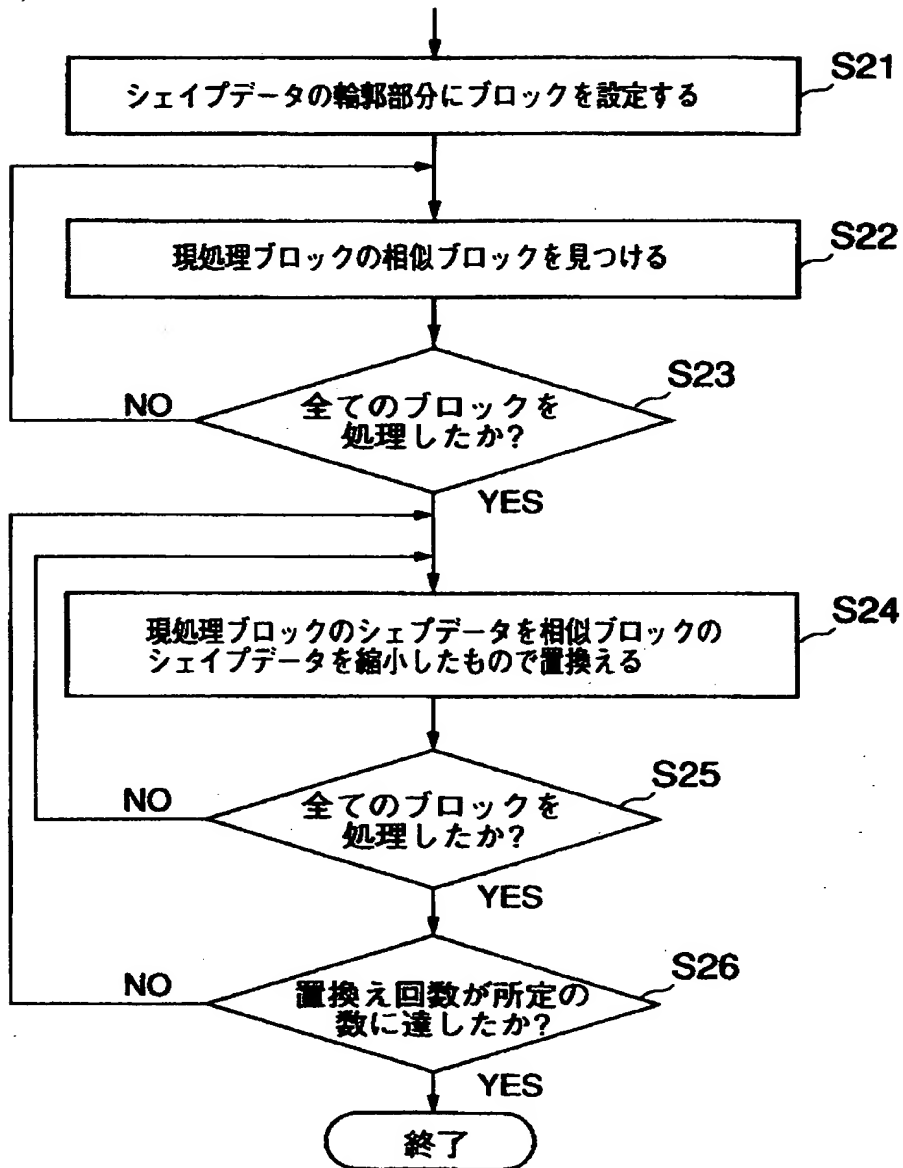
【書類名】

図面

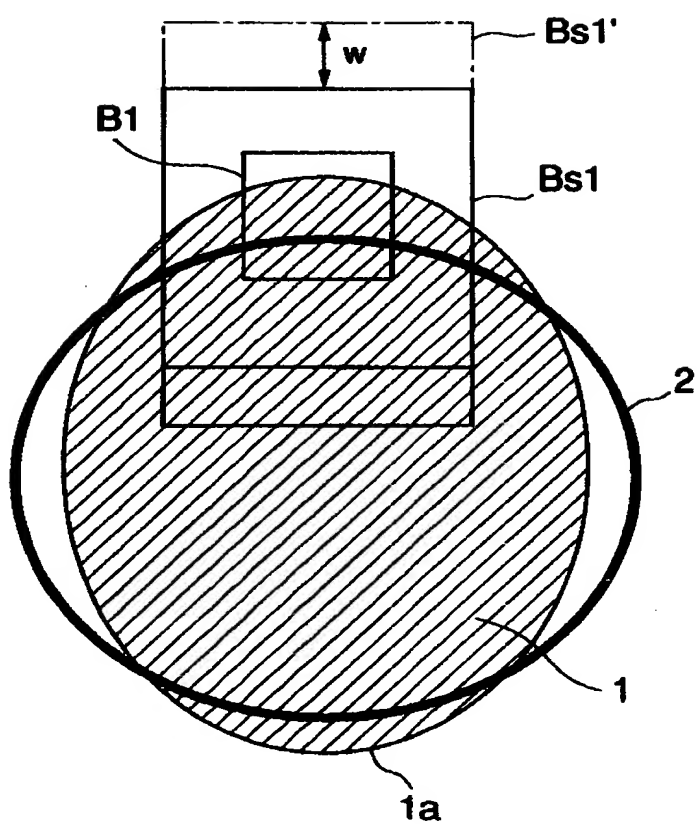
【図 1】



【図 2】

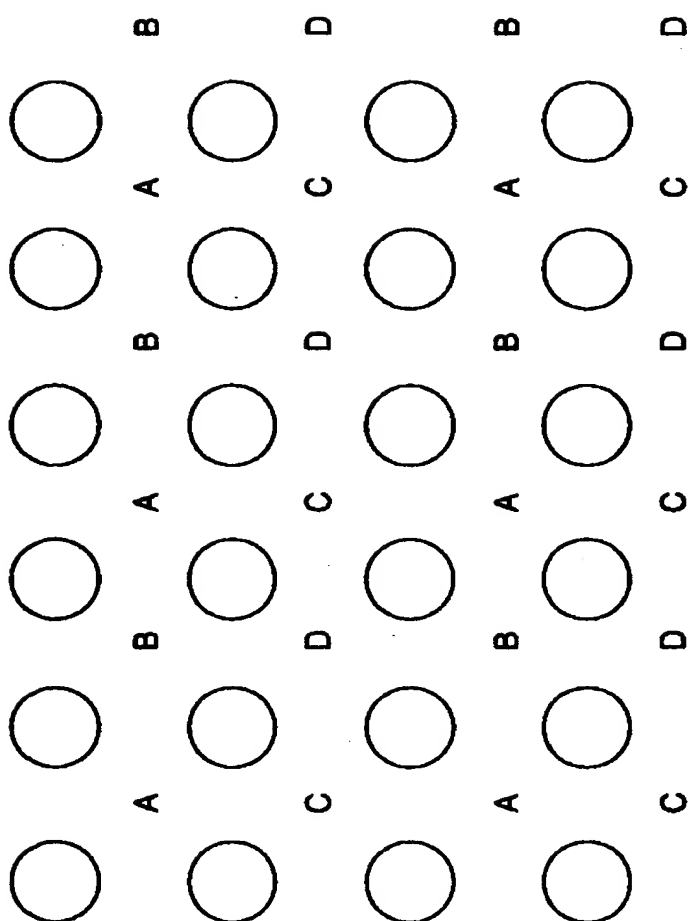


【図 3】

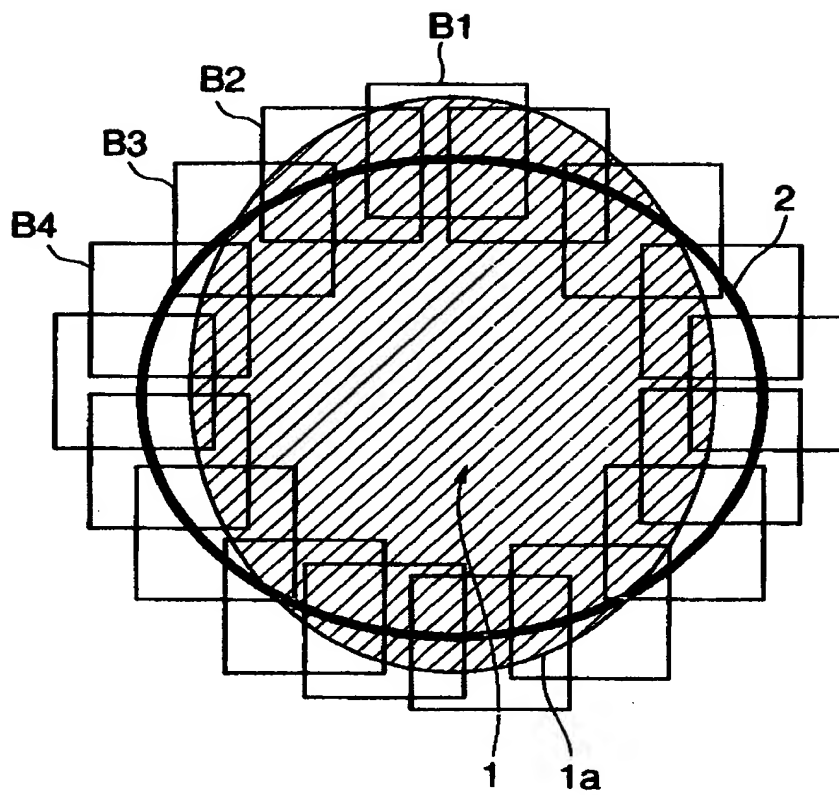




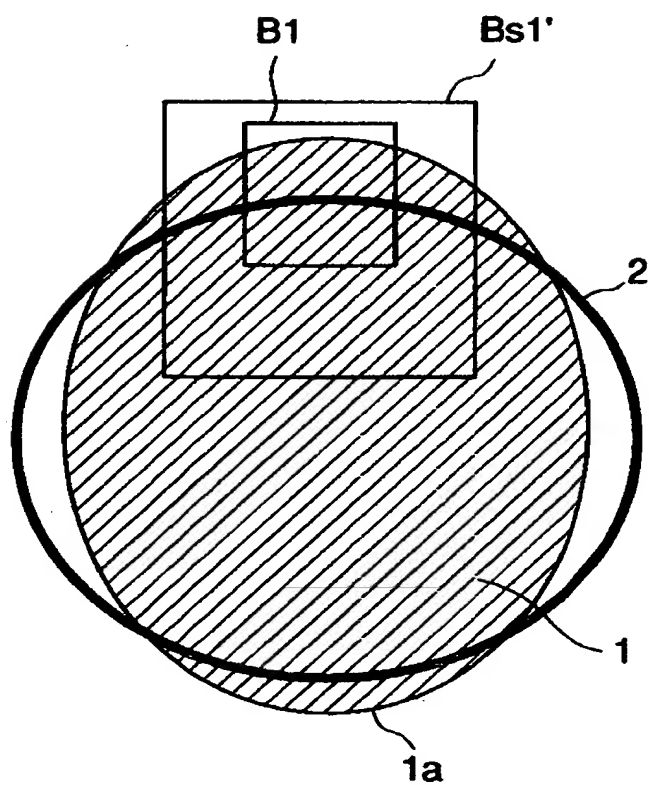
【図4】



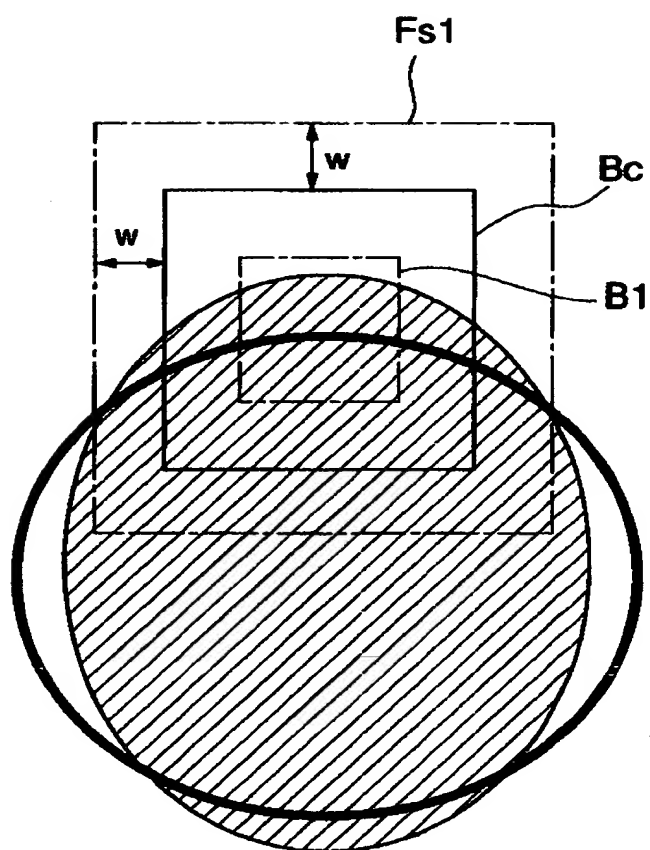
【図5】



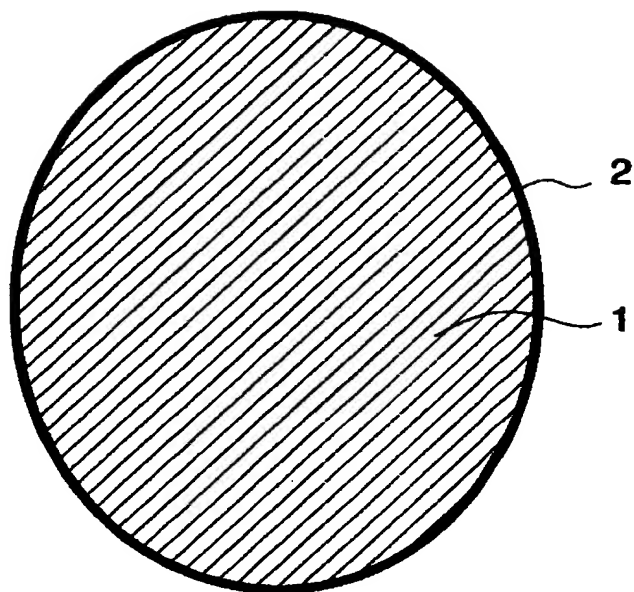
【図 6】



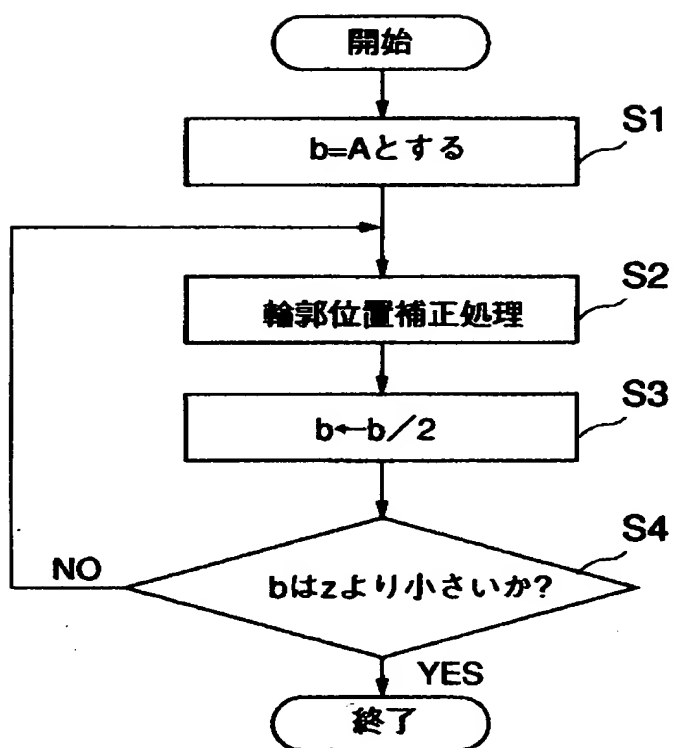
【図 7】



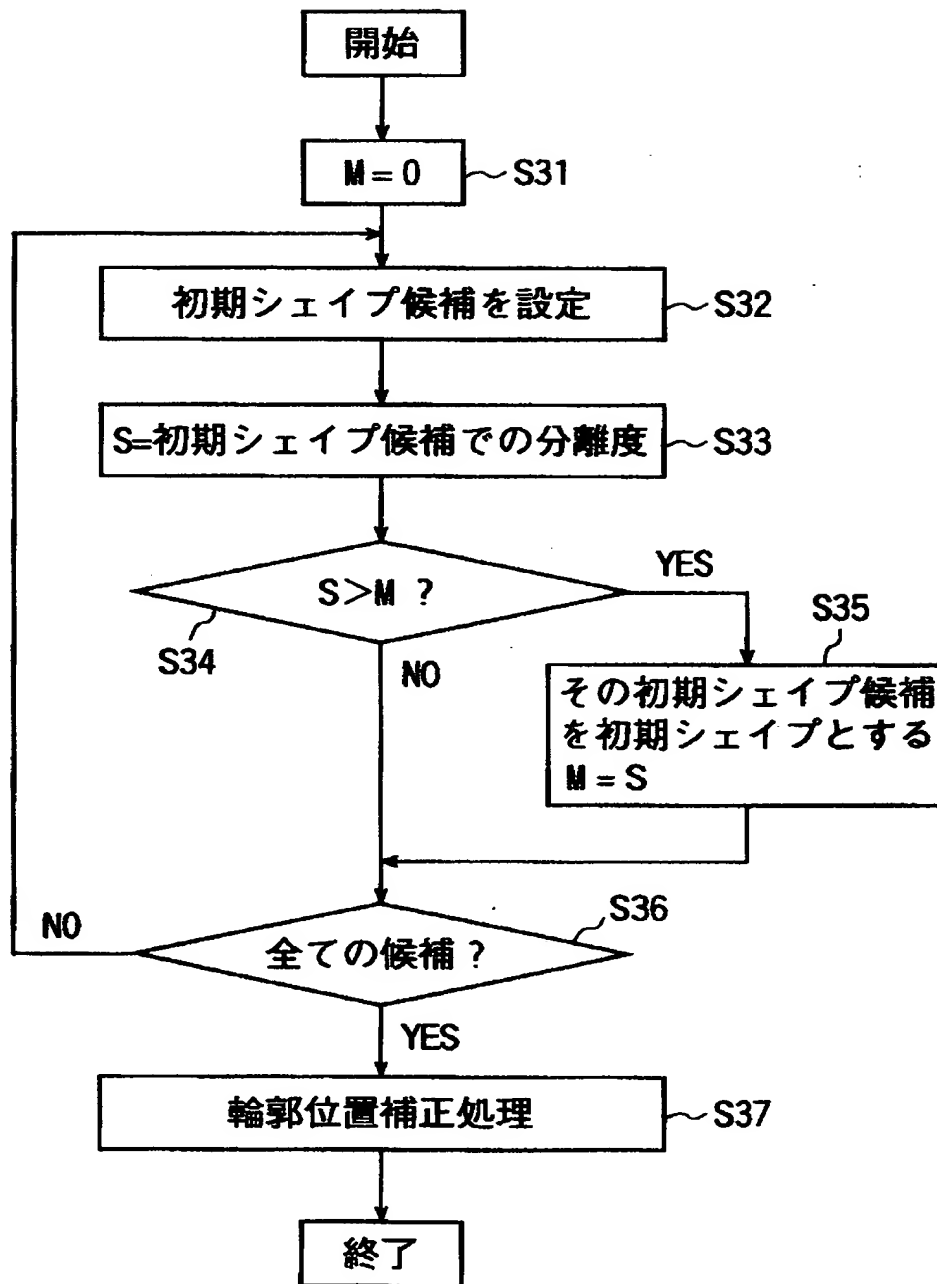
【図 8】



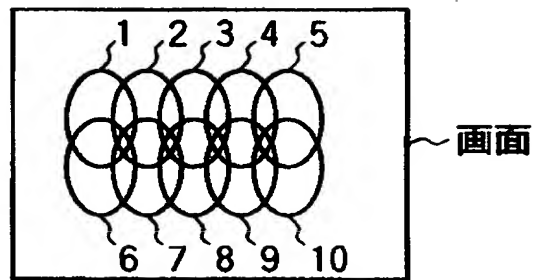
【図 9】



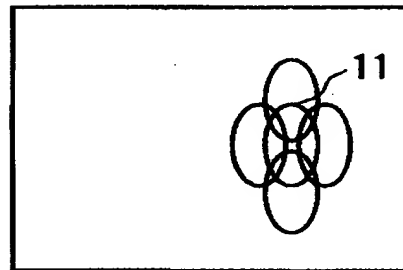
【図 1 0】



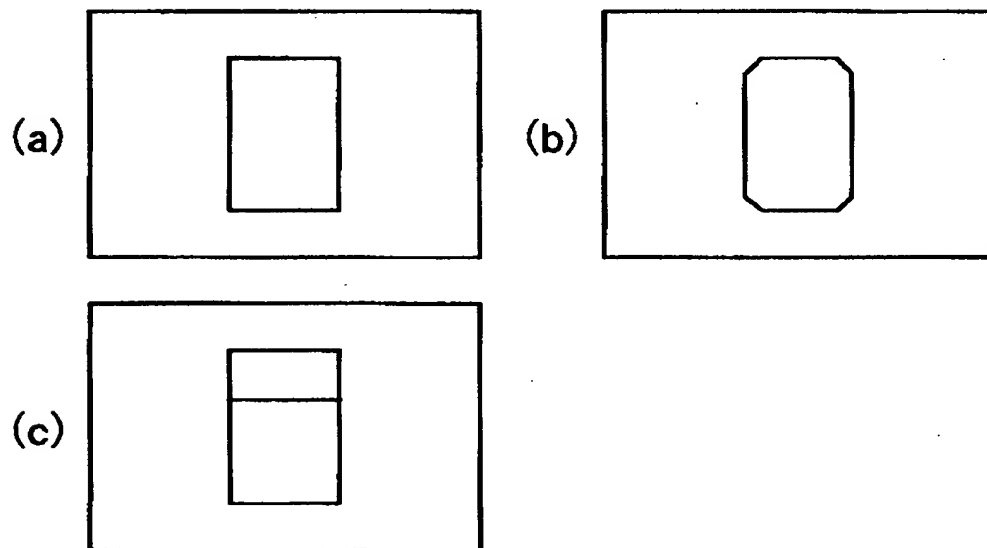
【図 1 1】



【図 1 2】

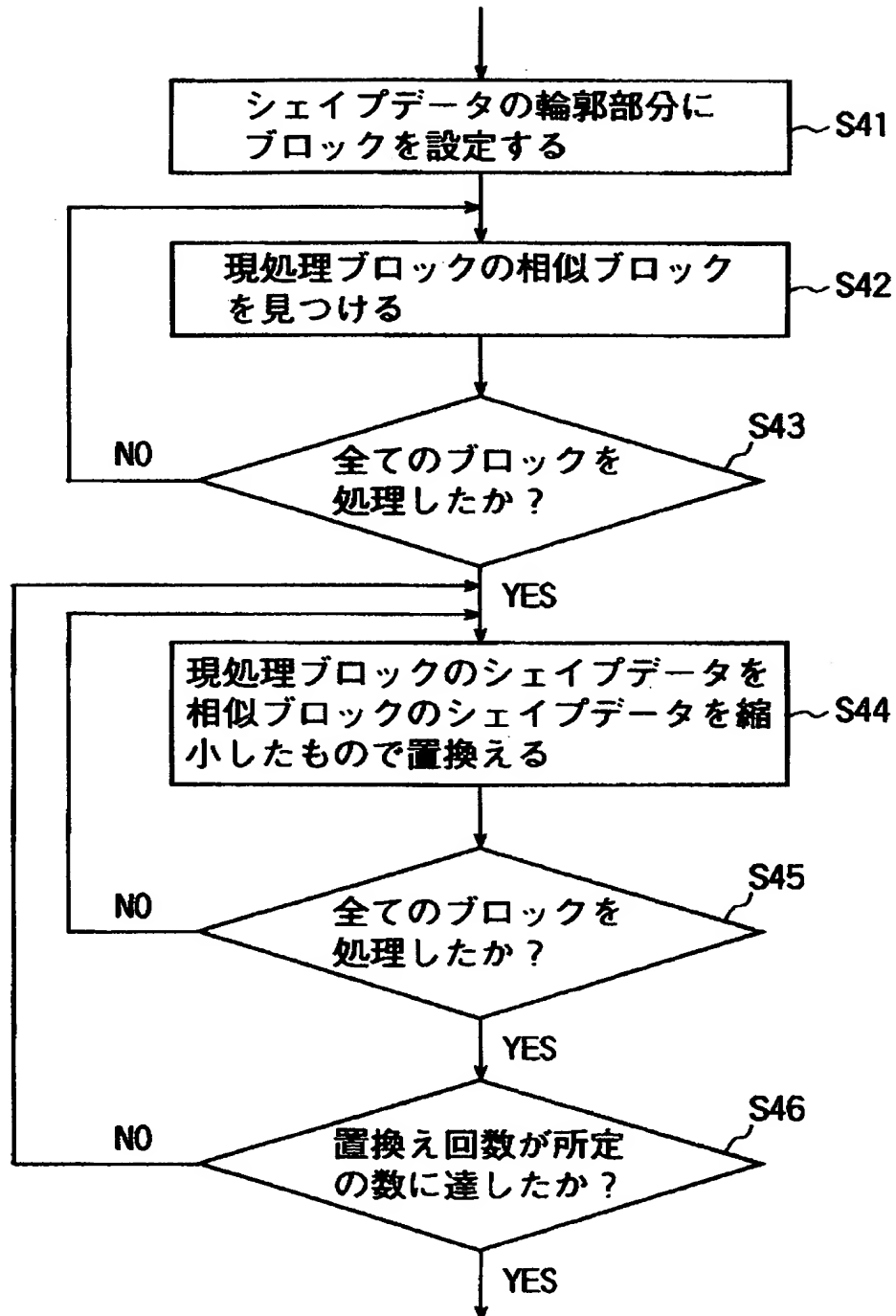


【図 1 3】

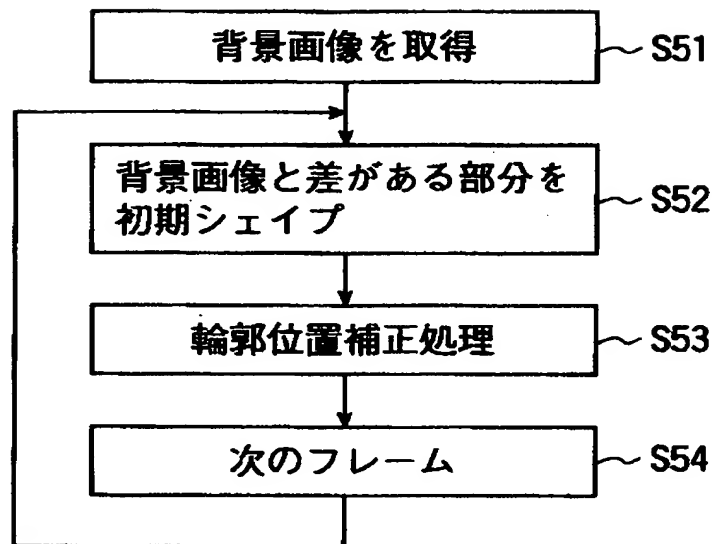




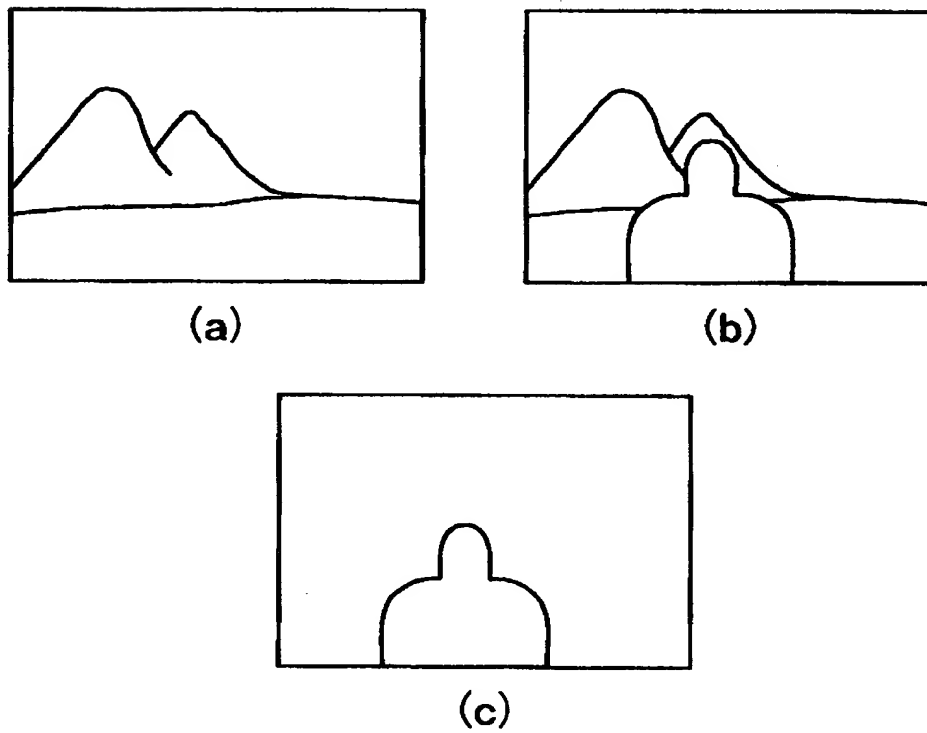
【図 14】



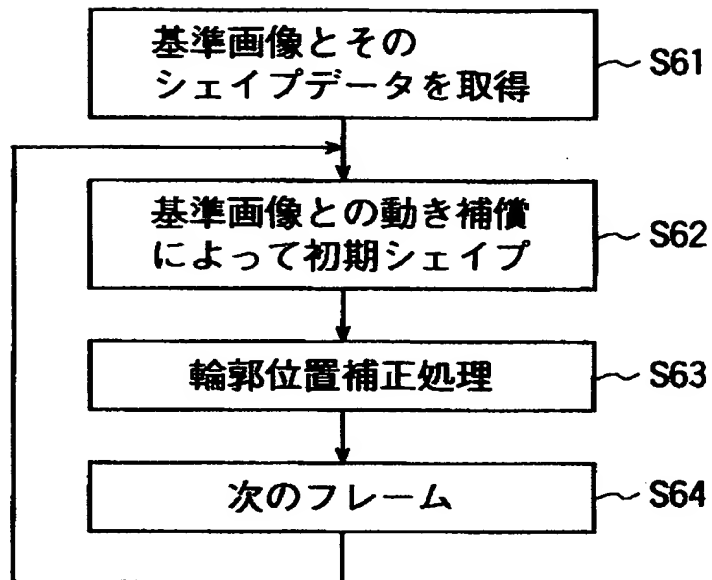
【図 15】



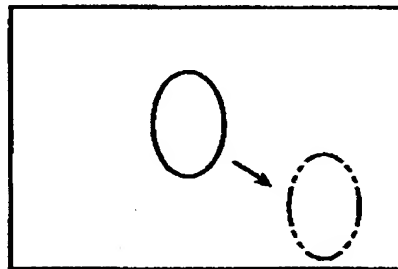
【図 16】



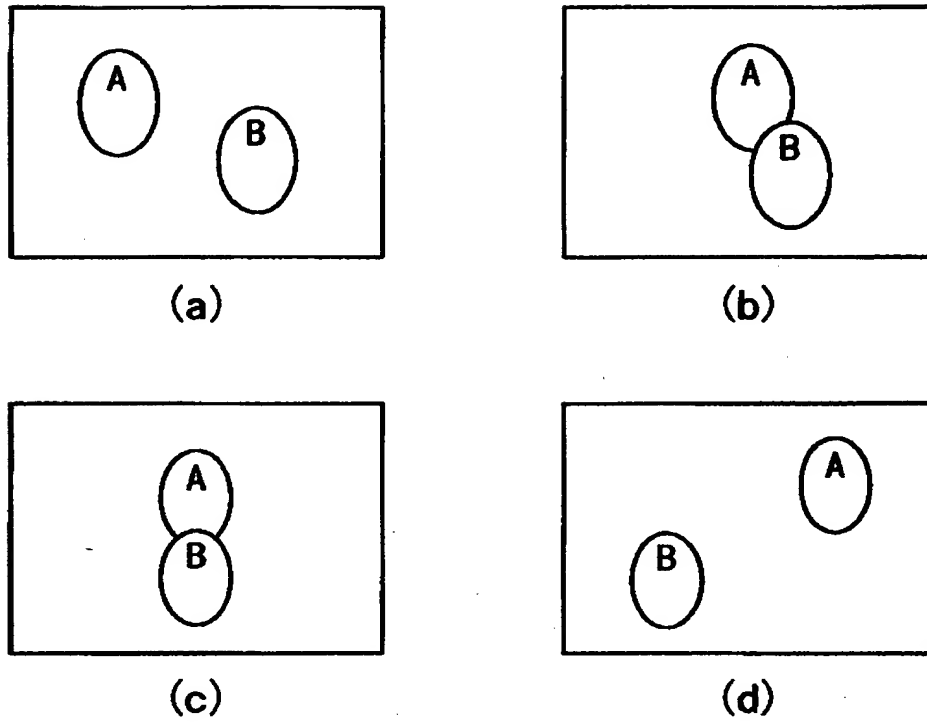
【図 1 7】



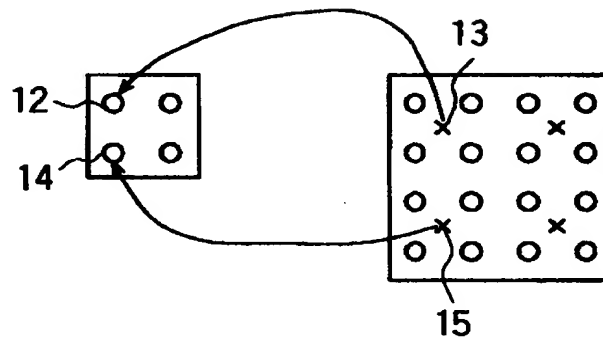
【図 1 8】



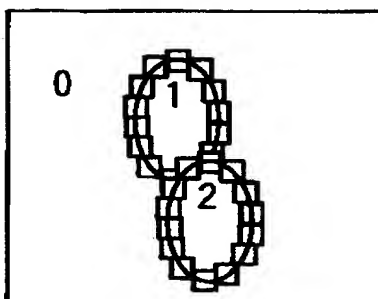
【図 19】



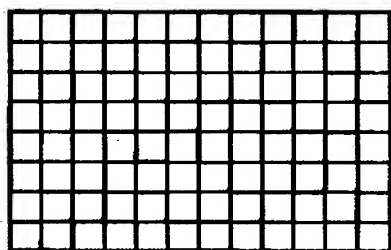
【図 20】



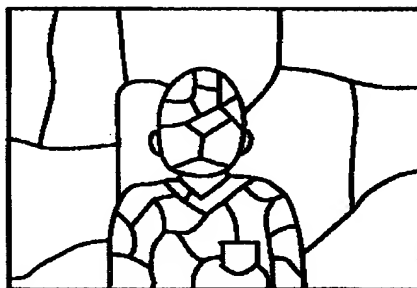
【図 2 1】



【図 2 2】

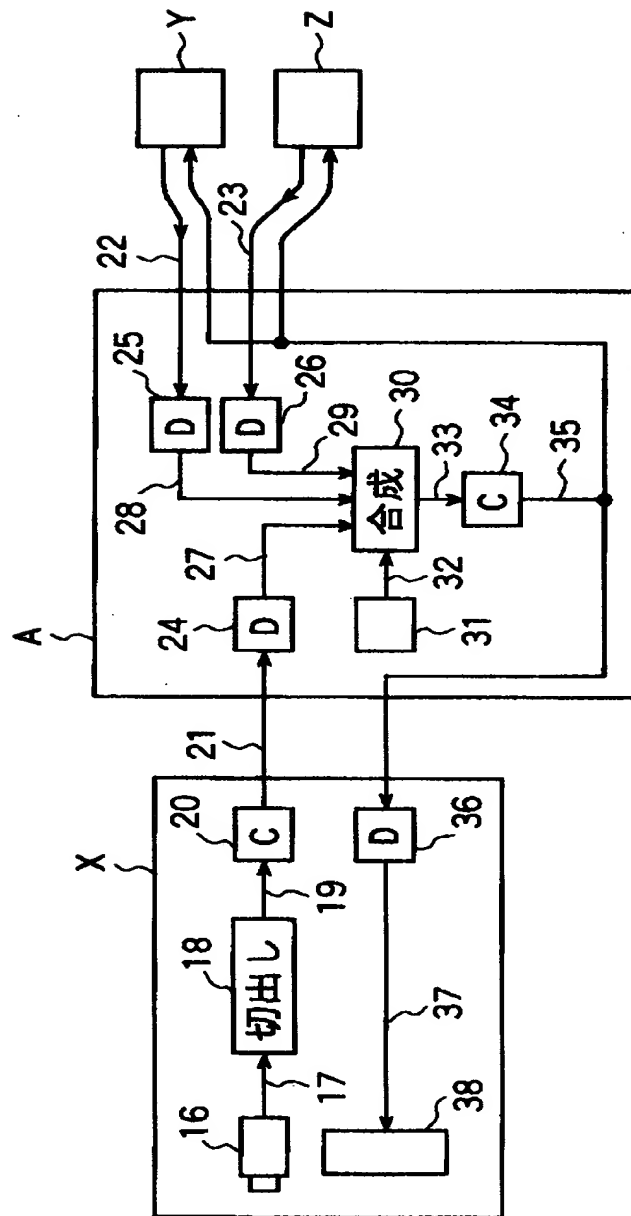


(a)

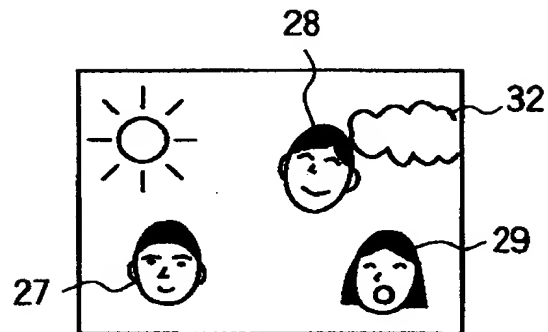


(b)

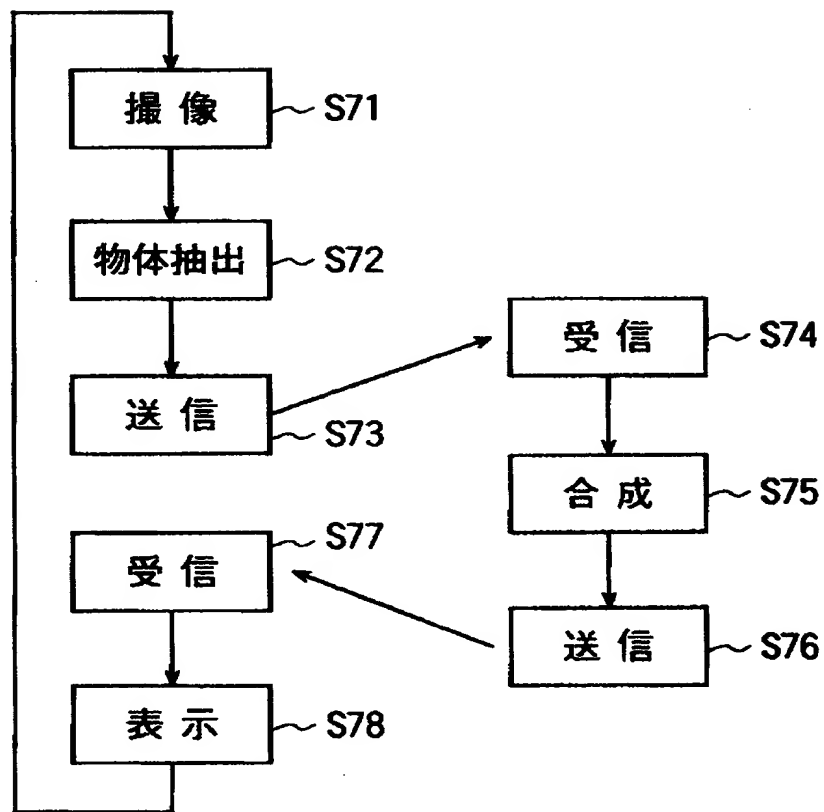
【図23】



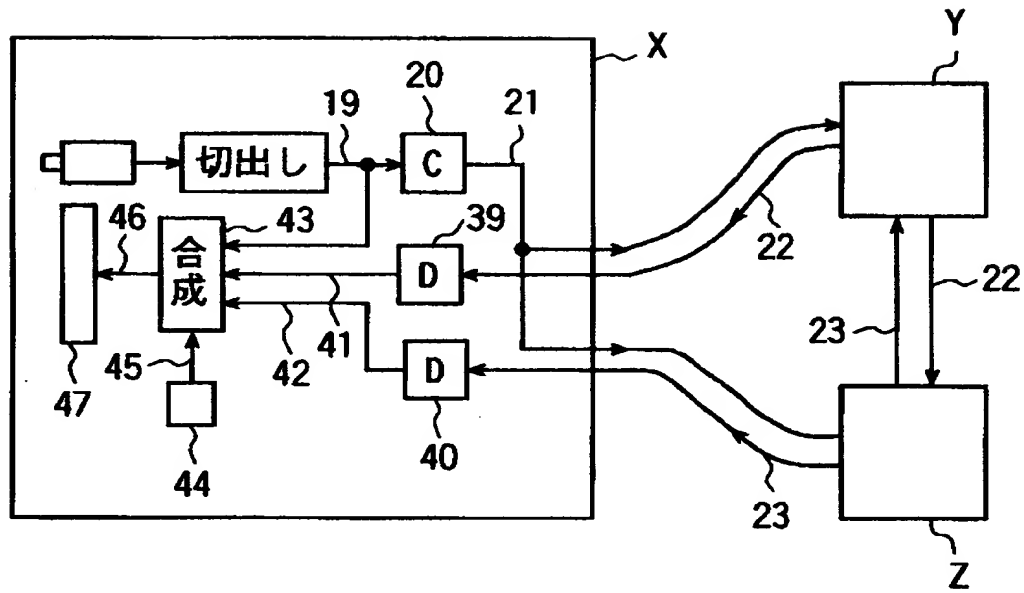
【図 24】



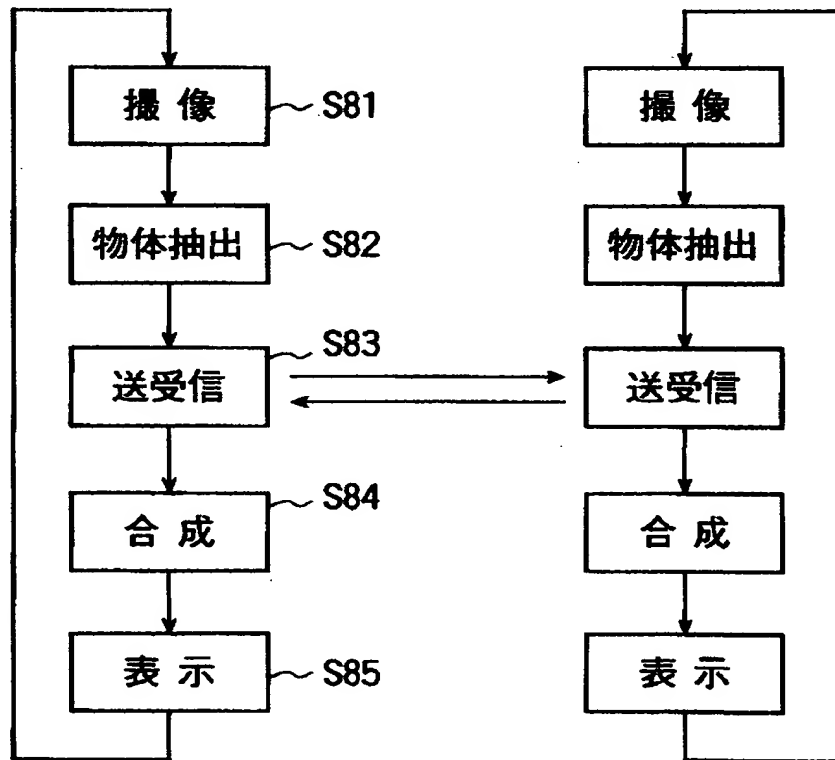
【図 25】



【図 26】



【図 27】





【書類名】 要約書

【要約】

【課題】 自己相似写像を用いる物体領域抽出方式の処理を高速にする。

【解決手段】 シェイプデータの輪郭部分に探索基準ブロックを設定する第1のステップと、各探索基準ブロック毎に画像データの図柄が相似であり、かつ、ブロックサイズがそれよりも大きい相似ブロックを同じ画像の中から見つける第2のステップと、各探索基準ブロック内の画像を二値化したデータを各々対応の相似ブロックのシェイプデータと置き換える第3のステップとから成り、第3のステップを所定の回数繰り返すことによりシェイプデータの輪郭線を前記物体の輪郭線に一致させる輪郭抽出処理において、輪郭抽出処理を反復して行い、その反復の初めには画像データとシェイプデータと探索基準ブロックは縮小して輪郭位置補正処理を行うようにする。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号

[000003078]

1. 変更年月日

1990年 8月22日

[変更理由]

新規登録

住 所

神奈川県川崎市幸区堀川町72番地

氏 名

株式会社東芝